WINDOWS: More Memory for Your Programs

DOS WORLD

The Essential Guide to DOS and Windows

July 1995 • No. 22

FREE Pull-Out File & Disk Command Guide

IT'S HOT

INSIDE OS/2 WARP

SPECIAL REPORT

How to Buy
Your Next PC

FREE Game: Madness!

How to Make Better Backups

New PC Checklist
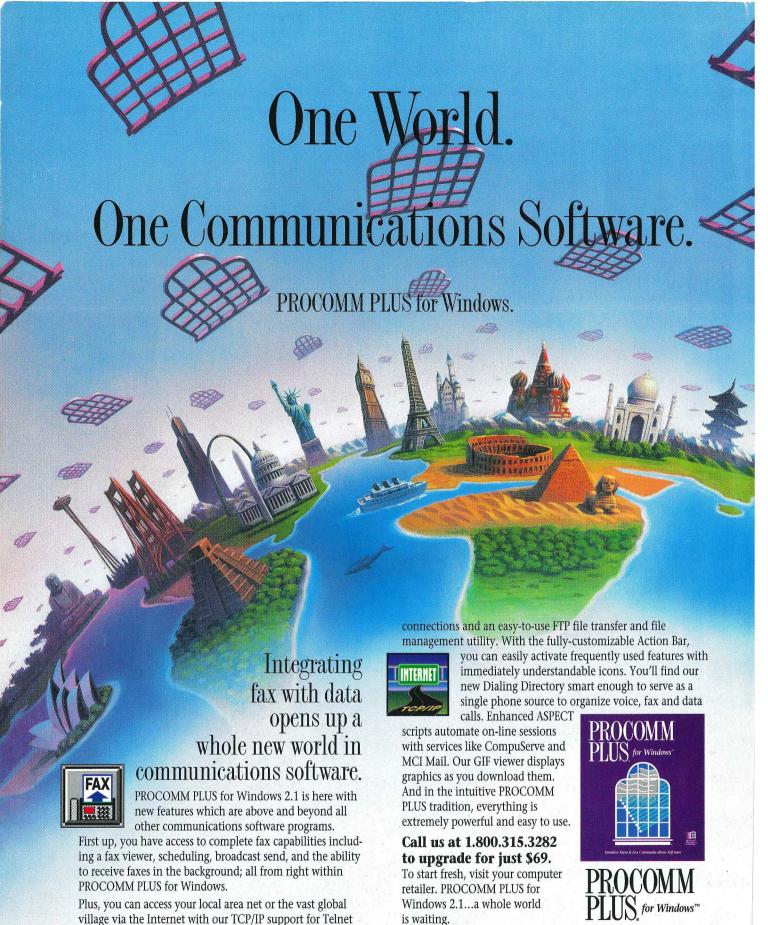- ☑ Monitor
- ☑ Bus
- ☑ CPU
- ☑ Hard Drive
- ☑ Peripherals

MICRO EXPRESS

Micro FLEX by Micro Express

MICRO EXPRESS

Batch File Special

Experts' Guide to Better Batch Files

PLUS: Batch File Math

US $5.95
Canada $6.95

07>

7 25274 78613 8

IS PC DOS 7 RIGHT FOR YOU? • Tips from Readers • DOS Watch • Q&A

Contents

FEATURES

DEPARTMENTS

MADNESS.BAS, page 53

All Roads Lead To the Superhighway

by Michael J. Comendul

Mick Jagger's on the Net. So's the White House. Most of the press is on, school children, and teachers. We've just joined the readers of a new magazine IDG is producing. Unfortunately, accessing the account requires more hardware muscle than many of our editors have currently. New 28.8kbps modems are on the way, but haven't arrived yet, so we have time to contemplate the Internet and what we'll be doing with access to this resource.

The Internet represents some of the most exciting territory on the frontier of computing. More than a territory (doesn't it strike you that writers spend too much time struggling for a metaphor to describe this thing?), it's a target that focuses the aim of PC technology. The Internet may represent the end of all those impassioned dreams about personal computing and its promise.

But not everyone agrees. Clifford Stoll, who, several years ago, wrote *The Cuckoo's Egg*—the nonfiction story (though it reads like a thriller) of his cracking an international spy ring traversing the Internet—has written a cerebral account of the Net and its negative impact. He calls it *Silicon Snake Oil* (Doubleday, $22). In it Stoll deflates the reputation of the Internet as a salve for our literary frustrations, our weakening educational system, and stress on the job, and debunks the Net as the answer to that most passé of habits—going to a public library. In fact, he looks at the Net as a symptom of social decay.

For corroboration he enlists support from someone who, for many people, represents the first line of defense in mankind's fight against the machine: Henry David Thoreau. Yet I wonder what Thoreau's opinion of the personal computer—or, rather, its use as a research tool for the Internet—might have been. He was a New Englander who embodied simplicity in his lifestyle, but, certainly, as his *Walden: Or Life in the Woods* proves, Thoreau's writing personifies self-reliance.

I have a theory about the Net's spreading influence, and I think people like Thoreau might agree. The Internet's power emerges in an era of national cynicism, where many people have given up hope that Washington, and the media, are telling the truth—about the economy, our national defense, our environment. The Net, like those live TV satellite feeds during the Gulf War, conveys a sense that it, too, is unedited and unfiltered. You access it, assimilate it, and draw your own conclusions.

Is that true? Is the Net unregulated? I don't know. I suspect it will be less true when major print publishers and entertainment producers get through with it. Just now, though, the Net has a valuable grassroots feel that even Thoreau might admire.

The Net is for those with a desire to find their own answers, yet I've experienced enough confusion, tedium, and literary detritus on the Net to realize that looking for worth and value out there is a full-time job. Around here, that's our signal to start a publication. We hope *Iway,* on sale in September, will offer a compass through the uncharted digital waters of the Net. Send us email at editors@iway.mv.com. ∎

Letters to the Editor

We're No Dummies!

Come on, Dan Gookin—give us some credit. The vast majority of your readers are experienced DOS users who have long since graduated from DOS 101. Your article, "The Seven Deadly DOS Commands" [*DOS World* #18, November 1994, page 17], was like a trip back to elementary school.

We use RD, DELTREE, and DEL just as often as we use MD, DIR, and COPY. These aren't commands that "everyone should avoid." If anything, inexperienced users should practice using them to become familiar with their potential risks.

Experienced DOS users read your magazine to brush up on little-known facts, tricks, and tips that can make difficult tasks easier and impossible tasks doable. Don't take us backwards by telling us what to avoid. Had this article appeared in *DOS for Dummies*, it might have served its purpose. Dan, we ain't no dummies.

Roland Curit
Virginia Beach, Virginia

I knew there was a pocket of hyper-intelligent DOS users out there. The people I normally deal with, through letters, e-mail, and questions on computer talk shows, really don't get DOS. They don't understand it; it confuses them. And all this time I assumed that they represented 85 percent of the people using DOS. That's why it's so nice to hear from you. I mean, no more will people in your neck of the woods use FDISK foolishly, zap directories accidentally with DELTREE, or use DEL *.* with wanton abandon. Heck, I may even write the MS-DOS group at Microsoft and tell them to stop putting in those annoying warnning messages for FORMAT C:. You've just shown me that my whole life's work—and three million copies of DOS for Dummies—has all been in vain. I congratulate you.

—Dan Gookin

Deep Background

I read [the following observation] with interest [in] the article "DOS and Windows: A Powerful Team" in *DOS World*'s Special Issue #5 [*Running DOS and Windows: Second Edition*, February 1995, page 6]: "Although you can run a DOS communications program from within Windows, you can't switch out of that program and work in another DOS application (multitask) while your communications software is still running. That means, for example, that you can't begin a long file transfer from an on-line service and then switch to another program within Windows."

I tested this statement by downloading a file and running two DOS applications in Windows at the same time. The only adjustment I had to make was to the PIF file that came with ProComm Plus 2.01 for DOS.

Under Advanced Multitasking Options, I changed Background Priority from 50 to 80. This let the file transfer work in the background while my other applications were up and I was working in one of them. (Also note that I set my DEFAULT.PIF and ProComm Plus's PIF to Full Screen. This uses fewer Windows resources.)

Paul S. Harrell
Columbia, Missouri

DOS Demand Remains Strong

Hey, Rich Freedman, whaddaya mean "there is no demand for a DOS uncoupled from Windows 95" [quoted in "DOS 7: Now You See It, Now You Don't," *DOS World* #20, March 1995, page 5]?

I'm a mechanical engineer whose boss hasn't yet increased the size of my hard disk to accommodate all my engineering programs, much less Windows, which eats up too much memory. In fact, there are 18 other computers in this company of stress analysts, and none of them has Windows on its hard disk.

I couldn't care less about using Windows. I'll do so only if I have no other choice. That time is probably fast approaching, with people like you making good working programs obsolete to sell new products. If it ain't broke, don't fix it!

Robert R. Allison
Huntington Beach, California

ERRORS AND OMISSIONS

Readers wishing to contact Bardon Data Systems, whose SmilerShell 2.1 was a featured shareware selection in *Running DOS and Windows: Second Edition* (page 85), can phone 510-526-8470 or send e-mail via CompuServe at **72340,375** or the Internet at **72340.375@compuserve.com**.

In the batch file TIMESUB.BAT, featured in "Tips from Readers," *DOS World* #20, March 1995 (page 13), the second line should read as follows:

```
IF %1X==CurrentX SET CURRTIME=%4
```

with *urrent* in lowercase, not upper.

Tips from Readers

Memory Savings

Many of your readers may know this, but perhaps some novice Windows users don't: If you always work in Windows and use only Windows applications, you don't need to add a DEVICE=MOUSE.SYS line to CONFIG.SYS, nor do you need to add a MOUSE.EXE or MOUSE.COM command to AUTOEXEC.BAT. Windows itself and most of today's mice come with a Windows mouse driver, and that's all you need. Windows, not DOS, handles the details of making mouse services available to Windows applications.

If you work only in Windows and your AUTOEXEC.BAT or CONFIG.SYS loads a mouse driver, you can save some memory by deleting the line that executes the driver or by preventing it from executing. To see how much memory—either conventional or upper memory (if you use LH to load the mouse driver high)—you can save, turn on your printer and type this command at the DOS prompt:

```
MEM /C > PRN
```

Look at your printout to see how your system is using memory. Then add a REM command (plus a space) to the beginning of the line that executes your DOS mouse driver. Save this change and restart your system. Finally, run MEM /C again and compare memory use.

Fletcher Hanks
FPO, New York

DEFRAG Protection Scheme

If you frequently delete files from a hard disk or floppy disk and don't want anyone to resurrect them with UNDELETE, try using a DEFRAG command to defragment your disk and, in the process, overwrite the deleted files.I use the following command to handle this task:

```
DEFRAG /F /H /S
```

The /F switch tells DEFRAG not to leave any empty spaces. /H says to move hidden files, and /S says to sort the files by size, with the smallest first.

If you run DEFRAG regularly, it shouldn't take very long to defragment a disk and overwrite sensitive information.

John Libby
Farmington, Maine

Start-Up Safety

I make frequent changes to my start-up files, CONFIG.SYS and AUTO-EXEC.BAT—either because I've added new software or in an attempt to optimize my system. To help me keep track of what lines I added when, I make a habit of printing these files whenever I update them. At the top of each printout, I write the date and note the reasons for the changes; then I store the printout with earlier versions of my start-up files.

If I delete one of these files accidentally, or a recently altered CONFIG.SYS or AUTOEXEC.BAT doesn't work properly, I have a record of start-up settings that do work.

Lane Olinghouse
Everett, Washington

Good advice, but we have one further suggestion: If you have DOS 6.x and you have trouble with your CONFIG.SYS file, reboot your system and press F8 when the message "Starting MS-DOS" appears on screen. This lets you execute the file line by line, which should help you pinpoint problems. If you have MS-DOS 6.2 or later, pressing F8 also lets you step through AUTOEXEC.BAT. If you have DOS 6.x, bypass both files by pressing F5. —*Eds.*

Speed Reading For Your CD-ROM

DOS 6.2 (and later) provides you with a way to read information from your CD-ROM drive faster, because its version of the Smart-Drive program caches read operations automatically.

There's just one catch: DOS's setup program always places the SMART-DRV.EXE command at the beginning of your AUTOEXEC.BAT file. As a result, SmartDrive doesn't know you have a CD-ROM drive and therefore doesn't cache its read operations.

Fortunately, fixing the problem is easy: Edit your AUTOEXEC.BAT file, placing the command that loads your CD-ROM driver—usually MSCDEX—immediately before the line that executes SmartDrive.

By the way, when SmartDrive loads, you can check the current cache status of all your drives by typing this at the DOS prompt:

```
SMARTDRV /S
```

Russell Stamets
Boulder, Colorado

Ask Your DOS: Haven't We Met Before?

by Hardin Brothers

How often have you looked at your DOS installation disks? Once you've installed DOS on your computer, you probably don't give them another thought. But, as the first question suggests, it may be time to examine those disks and their files more carefully. If you can't seem to read any files from the disks at all, you might want to start with the second question, which discusses failing disk drives.

Missing Links

The article "Sharing Resources with Inter-Link" in the January 1995 DOS World *[#19, page 23] sounds like exactly what I've been looking for. Although I have DOS 6.22 installed on my computers, the files* INTERSVR.EXE *and* INTERLNK.EXE *aren't in my system's* \DOS *directory. How do I get them?*

Burton Denman
Richardson, Texas

Get out your DOS installation disks. Make sure you have either an original installation set (which comes with new computers, whether DOS is preinstalled or not) or an upgrade set, which includes all files and manuals for a particular version of DOS.

In the \DOS directory on your hard disk, you should find a file called EXPAND.EXE. If it isn't there, copy it to your \DOS directory from the DOS Setup disk.

Next, check that first disk for a file called PACKING.LST, and view the file either with EDIT.COM or with another utility, or print it for future reference. PACKING.LST shows the name of each file on each disk, in both shipped form (which often ends with an underscore) and final form (which lacks the underscore). PACKING.LST tells you which disk contains INTERSVR.EX_ and INTERLNK.EX_. Put the appropriate disk into drive A (or B, if you'd rather) and type this line:

```
EXPAND A:INTERSVR.EXE A:INTERLNK.EXE C:\DOS
```

If you're using drive B instead of drive A, or if your \DOS directory is in a different location, modify that line accordingly. In a few seconds, you'll have the files on your hard drive.

The EXPAND utility decompresses files from Microsoft's distribution disks. It

> *If you can't find a standard executable file, or your system can't tell the difference between a 3.5- and a 5.25-inch drive, it's probably time to get reacquainted with your DOS installation disks.*

can't handle general compressed files, such as those created by PKZIP or LHARC. But it comes in handy whenever you need to load a new file from your DOS distribution disks or replace a damaged file.

You can use EXPAND in several ways. If you type just the name of the program from the command line, it prompts you for the name of the compressed file and the destination file's path and name. Or type this:

```
EXPAND A:INTERSVR.EX_ C:\DOS\INTERSERV.EXE
```

to tell it both names explicitly. The name you give for the first file can be either that of its compressed form, with the underscore, or the name of its decompressed counterpart. You can name several files and the destination storage area, as I did in the first example above. About the only thing you can't do is use wildcards in either the source or the destination filename.

You might save a few bytes of hard-disk space by deleting EXPAND once you've installed DOS. But make sure you can get it back if you need it. Eventually, you'll want to replace one of your DOS files.

Hardware Upgrades

I just upgraded my 386 computer with a 486 motherboard. Everything works great except for drive A, a 3.5-inch 1.44MB floppy drive. It used to work fine with the 386 system, but with the 486 motherboard, about 95 percent of my 3.5-inch disks don't work. I get the following error messages when I try to use them:

```
Sector not found
General Failure reading drive A
Data error reading drive A
```

I also tried the Windows File Manager and got similar error messages.

Jang Park
Fort Lee, New Jersey

It sounds as though you have a hardware rather than a software problem. There are several things you can do to try to fix the disk drive and, I hope, recover your data. The first and easiest is to clean the disk drive. You should clean drives regularly, to

make sure that the heads are in good shape. Because floppy-drive heads actually rub the surface of the disk (unlike the heads in hard drives), they pick up all sorts of contaminants that can make them fail. Most computer and office-supply stores stock small kits with a cleaning disk and a small bottle of cleaning fluid. The instructions are easy to follow, and a kit is a great investment in the future health of your computer.

If that doesn't solve the problem, the next thing to check is the CMOS Setup program. Make sure that each drive is defined correctly. Nothing will make a 3.5-inch drive fail faster than defining it as a 1.2MB, 5.25-inch drive.

Because you've just upgraded by adding a new motherboard, the third possibility is that you may have inadvertently created the problem yourself. Make sure that the drive-controller card, if you have one, is fully seated in its slot.

Also, make sure that the data and power cables are securely attached to the controller card or the motherboard and to the drives as well. A loose cable can cause all kinds of problems that are very difficult to diagnose.

In addition, consider the possibility that you need a new power supply. It's likely that your new motherboard uses more power than the old one. If the power supply isn't up to the job, or if you added another peripheral, such as a sound card, a CD-ROM drive, or a tape-backup unit, you may be taxing the power supply.

If the power supply isn't producing enough current, the disk drives could fail long before you notice any problems with the CPU or other internal parts of the computer.

If you decide to try a new power supply, make sure it's rated higher than your current one and that it's a compatible size. I've gone through the hassle of trying to squeeze a large power-supply box into a small computer chassis, and, believe me, it's not worth it.

Finally, you may simply need a new floppy drive. (Floppy drives have a much shorter lifespan than the computer itself.) But even changing drives may not make your old disks readable. If the old drive was failing, it may have recorded data incorrectly. If its heads were out of alignment, it may have put the data in the wrong place on the disks.

If your data is garbled because the old drive was failing, it may be impossible to recover no matter what you do. There are a couple of data-recovery companies around the country that use sophisticated equipment to read bad hard drives and floppy disks, but you won't want to pay their prices unless your data is extremely valuable to you or your company. ■

Technical Editor Hardin Brothers has been working with computers and writing about them for 15 years.

Best of the Batch

A Good Read

Have you ever tried to edit a read-only file with DOS's text editor, EDIT.COM? You can load the file and make all the changes you want, but there's a snag: DOS won't let you resave the file. You get a "Path/File access error" message, and your only option is to save the file under a different name. Wouldn't a "work-around" save a lot of aggravation here?

I've written a batch file called SMRTEDIT.BAT to solve this problem. (See the accompanying program listing, right.) It tells you whether the file is read-only and lets you edit it and save it, restoring the read-only attribute after you're done.

SMRTEDIT's syntax is the same as EDIT.COM's. For example, to load a file called REPORT.TXT, type the following line:

```
SMRTEDIT REPORT.TXT
```

If REPORT.TXT is a read-only file, SMRTEDIT.BAT gives you this message:

```
The file you've selected is "READ ONLY";
Do you still want to edit it [Y,N]?
```

If you press Y, SMRTEDIT.BAT removes the read-only attribute, runs EDIT.COM, and loads the file. When you quit EDIT.COM, DOS returns control to SMRTEDIT.BAT, which resets the attribute to read-only.

If you have a basic understanding of batch files, you'll figure out pretty quickly how SMRTEDIT.BAT works. The possible exceptions are the fifth and sixth lines, which determine whether the file is read-only.

The fifth line filters the output of the ATTRIB command through FIND to locate the read-only attribute (R). For example, if you type SMRTEDIT REPORT.TXT, DOS interprets the line like this:

```
ATTRIB REPORT.TXT | FIND "R  " > NUL
```

If FIND doesn't locate an R, DOS returns an ERROR-LEVEL of 1. The batch file captures this ERRORLEVEL in the sixth line and drops to the EDITIT subroutine, which loads EDIT.COM and the file REPORT.TXT.

If FIND does locate an R, then REPORT.TXT has a read-only attribute, and SMRTEDIT.BAT continues to the eigh-

SMRTEDIT.BAT lets you edit read-only files without resetting the file's attribute.

```
@ECHO OFF
IF %1!==/?! GOTO EDITIT
IF %1!==! GOTO EDITIT
IF NOT EXIST %1 GOTO EDITIT
ATTRIB %1 | FIND "R  " > NUL
IF ERRORLEVEL 1 GOTO EDITIT
CLS
ECHO.
ECHO.
ECHO.
ECHO.
ECHO.
ECHO.
ECHO.
ECHO.
ECHO.
ECHO.
ECHO     The file you've selected is "READ ONLY";
ECHO     Do you still want to edit it [Y,N]?
C:\DOS\CHOICE /N
IF ERRORLEVEL 2 GOTO QUIT
ATTRIB -R %1
C:\DOS\EDIT %1 %2 %3 %4 %5
ATTRIB -A +R %1
:QUIT
CLS
GOTO END
:EDITIT
C:\DOS\EDIT %1 %2 %3 %4 %5
:END
```

End

teenth line. The program subsequently tells you that the file is read-only and lets you decide whether you want to edit it anyway.

If you press Y, SMRTEDIT.BAT removes the read-only attribute, runs EDIT.COM, and loads the file. When you close EDIT.COM, SMRTEDIT.BAT returns the file to its read-only state.

As written, SMRTEDIT.BAT assumes that EDIT.COM is in the C:\DOS directory. In addition, it uses the DOS 6.x CHOICE command, so you can't run SMRTEDIT.BAT with earlier DOS versions.

Les Feinstein
Littleton, Colorado

Hints and Tips in DOS Programs

by Anne Fischer Lent and Stan Miastkowski

Whether you're a new user of the most popular DOS applications or an old hand, chances are you're not taking advantage of all your software can deliver. Every application has its neat tips, tricks, and shortcuts that can make your life easier, save keystrokes, and (best of all) increase your productivity. But finding these time-and-effort savers isn't always easy. Sometimes, hints and solutions are hidden in help files or buried in seldom-consulted manuals. More often, they're not documented at all. Here are 25 techniques that will help you get the most from your favorite DOS applications.

TIP #1 — Lotus 1-2-3 Release 4 for DOS: Wordy Headers

If you're developing multiple-page spreadsheets, a header (which prints on every page) will keep your audience focused on the meaning and significance of your well-honed numbers. But sometimes the standard one-line header in 1-2-3 spreadsheets just isn't enough. Here's how to add extra lines:

1. Choose Print, then Layout, then Title, then Header.
2. Type the text for the first line of the header.
3. Press Ctrl+A (a small triangle should appear in the command line), then the exclamation point.
4. Type the text for the next line of the header.

Note that you can repeat steps 3 and 4 to add still more lines to the header. Because 1-2-3 allocates only

three lines for the header, though, any lines you add beyond the third will overlap with data.

TIP #2 — Lotus 1-2-3 Release 4 for DOS: Going in Style

Lotus 1-2-3 has eight "named styles" that let you customize the look and feel (font, size, color, and so on) of all or part of your worksheets. For a spreadsheet that will wow those steely-eyed bean counters, follow these steps:

1. From the keyboard, choose Named:Styles.
2. Select the style you want to use.
3. At the "Change the attributes of range" prompt, select the cell or range of cells to which you want to apply the named style.
4. Press Enter.

1. To customize with a mouse, click the left button on the first cell in the upper left-hand corner of the range you want to format.
2. Drag the mouse to the end of the range you want to format.
3. Click on the Named Styles selector on the status bar.
4. From the pop-up list, click on named style you want to use.

TIP #3 — Lotus 1-2-3 Release 4 for DOS: Alternate-Cell Counting

How many times have you wanted to number a column sequentially (1, 2, 3, 4 . . .)? It's a common need in spreadsheets and databases and is usually easy to do. But what if you want to increment every *other* cell in a column? Here's how:
1. In cell A1, type 1.

Anne Fischer Lent and Stan Miastkowski have a combined total of 30 years' experience covering personal computers. They've written numerous articles on using computers in the real world and have coauthored two books, the most recent being The Windows for Workgroups Bible *(Addison-Wesley, 1993). You can reach them on the Internet at **alent@bix.com** and **stanm@bix.com**, respectively.*

2. Go to cell A4, type +A1+1, and press Enter.
3. Move to cell A4, select /Copy, and preselect the range in which you want to create the list.
4. For the FROM section, move the cell pointer down to A7 and press Enter.

TIP #4 Lotus 1-2-3 Release 4 for DOS: Graphic Moves

Placing graphs in your spreadsheet can certainly add to its effectiveness, but as your presentation evolves, you might want to move your graph elsewhere in the document. At first glance, it appears that 1-2-3 graphs are locked into the location where you created them, but there *is* a way to move them to another location in your spreadsheet or even to a different spreadsheet altogether. Here's how:

1. Select Graph, then Duplicate.
2. Select the graphic (or graphics) you want to duplicate.
3. Specify a range to which you want to copy the graphic (or graphics), either in the current spreadsheet or another spreadsheet altogether. Specify just the first cell in the range.

This procedure creates an exact duplicate of the original with a .1 extension. You can also copy a graph to a larger or a smaller range. Select Graph, then Settings, then Range.

Note that the graph you copy with Graph/Duplicate updates itself if you change data in the original. If you make changes to the original graph with Graph/Edit, though, the copied version of the graph doesn't change.

TIP #5 Freelance Graphics: Windows in DOS

You're on the road with a slick Freelance Graphics presentation, but your notebook doesn't do Windows. What now? Don't panic—you can run the screen show from DOS:

1. Select your screen show by choosing View/Screen Show/Prepare Standalone.
2. Specify a filename and path.
3. Click on the Options button, check the Run Screen Show Directly from DOS box, and click on OK twice to close each dialog box.
4. To run the screen show, type SHOW FILENAME.XXX and press Enter.

Note that if you're copying the screen show to the DOS machine, you should be sure to copy SHOW.EXE from the \FLW directory as well as the GX2 files created by Freelance Graphics for the slide show. Before you start the show, the GX2 files must be in the current \DOS directory.

TIP #6 Harvard Graphics for DOS: Going Mouseless

Although using any presentation-graphics program is easier with a mouse, some people prefer to stick with the venerable old keyboard. To navigate within the Draw portion of Harvard Graphics' version 3.x, follow these guidelines:

● To choose a menu item:

1. Use the spacebar or arrow keys to move to a menu item, then press F10 followed by Enter to select the item.
2. Type the number or letter that appears to the right of the menu item.
3. Press a speed key if one is available. For a list of speed keys press Ctrl+F1.

● To get or save a file:

1. At Filename, type the path rather than move up to the Directory field.
2. Leave two spaces at the end of the filename. (Harvard Graphics ignores any text following the spaces.)

TIP #7 Harvard Graphics for DOS: Print Extended ASCII

When you use extended ASCII characters in a graph with Harvard Graphics for DOS, you can print them by accessing the option "Use hardware fonts only from the output screen" (F8+Options menu). Set this option to No, so that your printer will use the Hardware Graphics soft fonts instead of hardware fonts.

TIP #8 ProComm Plus for DOS: Help Is on the Way

Like any powerful communications program, ProComm Plus for DOS offers a wide variety of options and commands. If you use the program, you're probably familiar with Alt+Z, which summarizes the software's commands. But did you know that ProComm Plus for DOS includes extensive context-sensitive help? (According to DataStorm's tech-support

staff, most users don't.) Here's how to delve deep into help topics:

1. At the terminal screen, press Alt+Z to get to the main help screen.
2. Press Alt+Z again to enter the context-sensitive help area.
3. Press T for a list of topics.
4. Choose your topic and press Enter.

TIP #9 ▶ ProComm Plus for DOS: Boot That Modem

If you're using an external modem and forget to turn it on before you start ProComm Plus for DOS, the program may appear to lock up at the start-up screen. (Normally, this happens only if your hardware flow control is turned on in the ProComm Plus for DOS set-up.) The program isn't really locked up, and there's no need to cold-boot your PC. Just press Ctrl+Break, then Esc a few times. That will bypass the start-up screen and get you into the Terminal screen. (Don't forget to turn on your modem.)

TIP #10 ▶ ProComm Plus for DOS: Modem Updates

If you have a brand-new high-speed modem, your stomach may be tied up in knots at the thought of trying to figure out the exact setup string ProComm Plus for DOS needs to send the modem. Don't despair—DataStorm regularly updates its modem drivers for all versions of ProComm to keep up with the avalanche of new modem models. You can get the drivers from the DataStorm BBS (314-875-0503). Once you're connected, go to the File Transfer menu and select item 1. Or, if you're a CompuServe member, type GO DATASTORM and follow the prompts.

TIP #11 ▶ Quattro Pro for DOS: A Quick What-If

When you want to compare the results of two different scenarios, use the Undo command to toggle between them. For example, raise the current fee scale by 10 percent but keep enrollment unchanged; press F5 (Undo) to return to your original scenario. Again, try increasing enrollment, but leave the fee scale unchanged; press F5 and you'll go back to the previous scenario.

TIP #12 ▶ Quicken for DOS: Fast Reports

Although one of Quicken's most-used features is its ability to generate highly detailed reports on the nooks and crannies of your personal or business finances, there are plenty of times when you want (or need) to generate a quick report on a particular section (your total salary so far this year, for example). These QuickReports are available in Bank, Cash, Liability, Asset, and Credit Card accounts. Here's how to access them:

1. In the register, highlight the field in the transaction for which you want a report (Salary, for example).
2. Press Alt+Z. The software displays a list with all transactions in the category you highlighted, with the total in the bottom right corner.
3. Press Ctrl+P if you want to print the list. If you want to access a specific transaction, highlight it and press F9, which goes directly to that transaction in the register.

TIP #13 ▶ Quicken for DOS: Instant Recall

If you have more than one memorized transaction with the same payee (two paycheck entries for different amounts, for example), you can move quickly from one to another:

1. Type the payee name. The first transaction to that payee is shown.
2. Press Ctrl++ (the Ctrl and plus-sign keys) to move to the next transaction for that particular payee. Continue pressing Ctrl++ until you find the transaction you want.
3. Press Enter to bring the transaction into the register.

TIP #14 ▶ Quicken for DOS: Fast Check Marking

When you're reconciling checks, one of the ways you can make the chore easier is to quickly mark a range of checks by number instead of laboriously accessing each check separately:

1. Display the reconciliation screen.
2. Press F8.
3. Enter the check-number range you want to reconcile (for example, 1045–1061).
4. Press Enter.

TIP #15 ▶ SPC Professional Write: Continuous Underlining

When you use Ctrl+U to underline, the spaces between words don't get underlined. This looks a bit unprofessional in the printed output. To underline those blank spaces, you need to create a "hard space": Hold down

Ctrl and press the spacebar. Using this hard space will place a small nonprinting dot on screen; the spaces between words will now be included when you press Ctrl+U to underline.

TIP #16 WordPerfect 6.0 for DOS: Speedy Printing

WordPerfect for DOS seems like a slug when it comes to printing, but there's a way to speed it up. To print documents faster from WordPerfect, select fonts that appear as "Built_in" in the Font dialog box. Don't choose a "Graphics Font," or printing will be much slower.

TIP #17 WordPerfect 6.0 for DOS: Automatic Spelling Correction

1. Select Tools/Writing Tools/Speller/Edit Supplemental Dictionary (or press Ctrl+F2+6).
2. Choose Add to place a new word in the dictionary.
3. Select Word/Phrase with Replacement to enter a common misspelling of a word, such as *teh* for *the*. (If there's more than one replacement, choose Word/Phrase with Alternates.)

TIP #18 WordPerfect 6.0 for DOS: Vertical Lines

1. To insert a vertical line between two columns, place the cursor in front of the first character in the first column.
2. Press Alt+F9+52131, Enter, Enter.
3. To insert additional lines between subsequent columns, just add a number to 5213; for a line between columns 2 and 3, substitute the number 52132 in the line in step 2.

TIP #19 XTree for DOS: Fast Archive Retrieval

XTree's built-in industry-standard ZIP and ARC file archiving is a tremendous help in saving disk room and tucking away little-used files into a space-saving format. But XTree contains an extra that does away with a major limitation of the plain-vanilla versions of these archivers: the inability to retrieve a single file (or a few files) from a large archive. XTree doesn't require you to decompress an entire archive just to get to the file you want. And it's a simple process, too. Here's how:

1. From the pull-down File menu, choose Open ZIP and ARC (or press Alt+F5 from the main screen).

2. To extract a single file from an archive, highlight it in the tree display and choose Extract from the File menu.
3. To extract several files, tag each one and press Ctrl+E.

TIP #20 XTree for DOS: Easy Comparisons

The longer you work with your PC, the more likely it is you'll end up with duplicate files scattered in various directories throughout your hard disk. XTree can help you clean up the mess with fast comparisons:

1. From the Alt File Commands window, press F4 to access the COMPARE command.
2. A submenu appears, letting you choose from six options:

● Duplicate Names, listing all files without unique names
● Unique Names, listing all files with unique names
● Identical Names, listing each file with the same name, date, and time as another file
● Newest Date, listing the most-recent version of duplicate files
● Oldest Date, listing the earliest version of duplicate files
● Scope, letting you choose whether to compare all files in one volume, all files across all volumes, or files with matching paths

3. Make your choice from the submenu. After the list of duplicate files appears, use standard XTree commands to delete, copy, move, or perform other common functions on individual files or groups of files.

TIP #21 XyWrite for DOS: Close at Hand

Run XyWrite for DOS under Windows and place its icon in the same group with your Windows word processor. When you need to edit or prepare ASCII text, you can pop over to XyWrite without leaving Windows.

TIP #22 XyWrite for DOS: Maximum Memory

Chances are you don't need to use every feature of XyWrite all the time. So why waste precious memory? To free up some of the memory stolen away by XyWrite modules you don't need, such as the spell checker, use the UNLOAD command at the command line. When you try to use the spell checker again after you've unloaded it, XyWrite automatically reloads the function for you. ■

Pennies From Heaven

by Hardin Brothers

I f this month's nominations are any indication, the shareware industry is changing in two important ways. First, the cost of registration is going down. This month's programs are priced at a low $8 and a very reasonable $20, respectively.

But low cost, by itself, isn't enough to get users to try shareware and eventually register it. Users pay only for first-rate programs. The second trend I'm seeing is that shareware quality is definitely on the upswing.

This month, we feature a pair of programs many of you will want to use, keep, and register. Whether you're a corporate or home-office worker who needs to keep track of your time for billing clients, or a programmer or administrator who needs to make disk copies fast, these utilities offer quality and value.

The programs discussed here are available from the *DOS World* bulletin-board system (603-924-3181), as well as from most information services and local BBSes. (See the accompanying sidebar, "Share the Wealth," page 22, and the *"DOS World* BBS" section of "How to Use This Magazine," page 64, for details on accessing the bulletin board. Prices are listed in the "Product Information" box, page 22.)

Time Is Money

I'd like to nominate the shareware program LOG for review. It records computer usage interactively, from the command line, or via batch files. You can view or edit "project" files from within the program, with total usage available at any time. If you charge clients for your time, you can use LOG to get an accurate record of billable hours. It can also determine how much time you spend using your favorite apps and games.
Douglas Park
Biloxi, Mississippi

I've been looking for a program like LOG. And I'm particularly impressed with its ease of use; LOG needs only a scant three and a half pages of documentation.

When you first start LOG, you set up the accounts or projects you want to track. For example, I started one called SWARE to track the time I spend on this shareware column. (Because you use the account name as a filename as well, you're restricted to eight characters.) Once you define the account name, the program asks you to enter a short description, such as "Reviewing DOS Shareware." And that's really all there is to creating a new account or project.

LOG uses the information to set up a database of all your projects. It creates one file for each project in the database. When you want to clock into a project, you can do so interactively or simply type LOG IN SWARE. When you're ready to clock out, type LOG OUT. If you clock into a second project, the program assumes that you want to clock out of the first.

At any time, you can review the number of hours you've spent on each project. You can sort the display by time spent on each project, by project name, or by project creation date.

Unlike some commercial time-tracking programs, LOG doesn't get in your way. It assumes that you want to turn the clock on and off for various proj-

> *Even if you're short on cash, you can get business and personal-productivity software that's out of this world.*

SHARE THE WEALTH

Our "Shareware Exchange" column invites you to send a copy of your favorite shareware program, along with a description and an explanation of why you like it (500 words or less) to Shareware Editor, *DOS World*, 86 Elm Street, Peterborough, NH 03458. Tell us how we can obtain a copy of the program. We'll pay contributors $50 for each program featured in this column. Please don't send us shareware you've written yourself—we'd prefer recommendations from users, not authors.

You can contact us on CompuServe at **75300,2361** or on the DOS World BBS at **603-924-3181**. All programs featured in "Shareware Exchange" are available electronically from the DOS World BBS (instructions on page 64). LOG (LOG11.ZIP) and Disk Copy Fast (DCF49.ZIP) are in File Area 5. In addition, most of the programs described in "Shareware Exchange" can be found on major on-line services and local BBSes.

—Eds.

ects instead of guessing based on the active application. You can have it track how long you spend in various programs by writing batch files to start LOG, start the programs, and then stop LOG when you're done.

LOG can't produce printed reports (except for screen dumps) or generate bills. But you can save a lot of money by registering LOG instead of buying a commercial time-tracking program, and then using your word processor or spreadsheet to prepare invoices.

LOG costs only $8 to register. If you have to keep track of your time, or if you're simply curious about where your time goes each day, that's a bargain.

Copycat DOS

*I'd like to suggest one of my favorite shareware programs: **Disk Copy Fast** by Chang Ping Lee. It's a single-pass copy program that's not only easy to use, but fast. It copies, compares, and formats a disk in one pass; it features an easy-to-use window. I also use this program when I want to format disks; it just reads a blank formatted disk and then writes to the unformatted disk. DCF is faster than any other format or disk-copy program I've tried.*

William L. McCarter
Paragould, Arkansas

When I tried Disk Copy Fast version 4.8, I was severely disappointed. The documentation sounded great, and I was eager to try copying a few disks. But DCF refused to read any disks on my computer, even though DOS, Windows, and a handful of utilities had no problem at all.

I was about to forget about DCF and move on to another program. Then one day I was snooping around SimTel, a big shareware repository on the Internet, and I came across DCF version 4.9. I downloaded and unzipped it, gave it a try, and it worked beautifully.

DCF is relatively fast. (Floppy disks will never be truly fast, I'm afraid.) It's smart enough to read only the parts of your source disk that actually contain data. It stores data in memory and waits for a disk to which to copy it.

During the copy, you can choose to have DCF format the destination disk as it's copying, and to read every byte back to make sure the data is the same as written, which is a much safer (but slower) check than that performed by DOS's VERIFY or COPY /V command.

One way DCF increases disk write speed is by timing its write commands so that the disk never has to make a second revolution to get to the next sector. Another part of its speed is simply a result of being smart about which sectors it needs to write at all.

In any case, DCF 4.9 is a useful utility if you need to copy disks fairly often. Both noncommercial and commercial versions are available. The former can make only ten copies of a source disk at a time; the latter (which costs twice as much) can make unlimited copies of a source disk.

The same author also offers several other disk utilities, all of which are described in the Disk Copy Fast documentation. ■

Technical Editor Hardin Brothers has been working with computers and writing about them for 15 years.

PRODUCT INFORMATION

Prices listed are for registration only. Vendors may charge additional small fees for shipping and handling, extra disk copies, and printed documentation.

Disk Copy Fast 4.9, $20
DCF Software
P.O. Box 60064
Palo Alto, CA 94306

LOG 1.1, $8
Dr. Brian Sutton
7506 Ehrlich Road
Tampa, FL 33625
813-920-4044

SAFEGUARDING YOUR DATA

A Backup Blueprint

If we all know how important it is to back up our files, why do so many of us put it off? The solution is to find a routine. Here's a plan to make backing up less of a chore.

by Ken Johnson

In the days of the 20MB hard disk, backing up your entire system onto floppies wasn't too difficult or time-consuming. But with hard disks today pushing 500MB, backing up with anything but a tape drive seems absurd. The thought of sitting in front of a PC swapping disks for a 1GB hard-drive backup is enough to make procrastinators of us all.

But you *can* safeguard your files without purchasing additional hardware by developing a simple backup strategy using the tools within DOS itself. (See the sidebar "Seven Steps to Safer Backups," page 24, for a quick summary of the tips and advice discussed here.)

The foundation of a sound backup strategy is to know what to back up and what *not* to. Obviously, you want to safeguard important files that change frequently,

Contributing Editor Ken Johnson is training and support manager at the law firm of Mayer, Brown & Platt in Chicago. He's a columnist and writer for several business and computer-law publications and serves as a contributing editor for the American Bar Association's NETWORK 2d *newsletter.*

such as data files, text files, configuration files, program source code, and batch files.

What you don't need to back up on an ongoing basis are executable program files (with the extensions EXE, COM, DLL, OVR, and so on), because they don't change. You need only two backups of program files—the original program disks and, just in case, copies of those disks.

Besides programs, you also don't need to back up "junk" files left over from other programs or operations—things such as backup files (with the extension BAK), temporary files (TMP and $$$), Windows temporary files (~*.*), and recovered lost-cluster files, which are named FILE*.CHK in the root directory. (In fact, you should delete most of these. For more information on streamlining your hard disk, see "Managing the Maze," *DOS World* #21, May 1995, page 24.)

If you're using MS-DOS 6.x's UNDELETE with the Delete Sentry option, don't back up the hidden \SENTRY subdirectory. That's where Delete Sentry moves deleted files so that your system can recover them if necessary. When you delete

a large number of files, the \SENTRY directory can become quite large, up to a maximum of 20 percent of the space on the disk.

To help with your backup strategy, organize your hard drive to keep data files in separate subdirectories from program files. That way you can do a simple backup of entire subdirectories, without selecting data files individually within them.

Many programs create their own data directories when you install them, but some don't, dumping everything into the program directory. In that case, set up the data directory manually and point the program to it.

DOS's Own Backup Tools

DOS has included a BACKUP program since version 2.0, but it has been notorious for lacking basic features such as easy file selection, file exclusion, and data compression. Recreating files required a separate program, RESTORE. In addition, Microsoft changed BACKUP in various versions of DOS, making it incompatible with data backed up in previous versions. For example, BACKUP in DOS 3.3 couldn't restore files backed up in DOS 3.2.

In MS-DOS 6.0 Microsoft introduced a new, "friendlier" backup program called MSBACKUP (a scaled-down version of Symantec's Norton Backup program). MSBACKUP includes a graphical interface, integrated backup and restoration (there's no MSRESTORE program), reusable settings, multiple backup types, and data compression.

Unfortunately, limitations still exist. MSBACKUP doesn't restore data backed up by previous DOS versions, and the program is somewhat cumbersome, with multiple layers of menus. (See the sidebar "A Switch in Time," opposite, for tips on automating MSBACKUP.)

Choose Your Weapon

Most of the programs we're talking about, including MSBACKUP, offer three backup types: full, incremental, and differential.

A "full" backup doesn't necessarily mean backing up every file on your computer. A full backup copies all files you've selected, whether or not they've been modified since your last backup.

Full backups are the starting point of your backup strategy. Between full backups you need to do only incremental or differential backups. Both are partial backups that select only files that have been modified or to which material has been added. (When a file is modified, created, or copied to a new location, DOS turns on its archive attribute.)

Differential backups select files that have been modified, but don't turn off the archive attribute. A new differential backup copies the same files, along with any newly created or modified files. Thus the last differential backup will include all changes since the last full backup; the full backup and this differential backup give you the complete set of selected files you might need to restore.

Incremental backups select files that have been modified, then turn off the archive attribute. An incremental backup, then, handles only files created or modified since the last full or incremental backup. To maintain a complete set of files, you must keep the full backup plus each subsequent incremental backup. If you have to recreate files, you must restore each incremental backup in the order in which it was made.

Differential and incremental backups have their own advantages and disadvantages. Differential backups let you use the same disks, but take longer as the backup cycle goes on. A differential backup is generally the fastest method if you work with the same group of files each day. Incremental backups are the fastest choice if you work on many different files every day; unless you're using XCOPY for your backup plan, though, incremental backups require new disks each time. Incremental backups, unlike differentials, also let you recover previous versions of the same file.

Whatever method you choose, when the differential backup becomes too long or you end up with too many incremental backup disks, it's time to start the cycle over with another full backup. Just make sure to pick one backup method and stick with it; mixing incremental and differential backups may mean that you won't be able to restore some files.

Automating Your Technique

The best strategy is to make backups part of your everyday work at the computer. If you make backups easy, you'll do them automatically —instead of putting them off. You can do routine backups with a special batch file or by using backup logic in your other batch files.

And don't think you have to use only MSBACKUP, because other commands and programs can perform the backup function. Consider XCOPY. XCOPY with the /A switch does a differential "backup": It copies all files with the archive attribute, but doesn't turn the

SEVEN STEPS TO SAFER BACKUPS

A backup strategy involves not only what to back up and when, but how to prepare for the backup and what to do afterwards. Here are some rules for safeguarding your hard-disk data:

1. Determine which files you need to back up and which you don't.

2. Run SCANDISK or CHKDSK before a major backup to look for file or disk problems. Correct any errors it finds before you run your backup program.

3. Develop a backup cycle. A cycle begins with a full backup of selected files. Then do incremental or differential backups of the new and modified files, using the archive attribute as the selection criterion. Do a full backup again on a regular schedule (say, once a month), or when the differential or incremental backups get too big.

4. Don't mix incremental and differential backups. Doing so may result in an incomplete set of files if you have to do a restoration.

5. Rotate two sets of backup disks (or tapes) for security. Label them carefully with the set and disk number. Restoration programs require you to insert disks in the order in which you created the backups. Without proper labels, you'll do a lot of disk shuffling.

6. For that extra measure of safety, don't keep backups all in one place. Some people keep backups in a fireproof vault or in a separate physical location from the PC. This protects against both theft and natural disaster. For extra protection, make a copy of the backup disks and store the duplicates in different places.

7. A backup strategy won't help you at all if you don't stick to it. Automate your backup process as much as possible so that it won't seem like a chore, but part of your daily routine.

—K.J.

archive attribute off after the copy. XCOPY with the /M switch does an incremental "backup." It copies files with the archive attribute on, and turns it off after each copy.

If you use batch files to run your applications, consider adding XCOPY backup logic to copy that application's data files to floppy. If you have more files than can fit on a disk, the batch file can handle it if you're doing an incremental (/M) backup. XCOPY returns an ERROR-LEVEL of 4 when the disk fills, so use an IF statement to check for that, pause to swap disks, and re-execute the XCOPY command. Files already copied won't be copied again. (See the sidebar "A Switch in Time," right, for more.)

My personal backup strategy uses still another alternative to MSBACKUP—the shareware compression program PKZIP. PKZIP compresses multiple files into one large ZIP file, which can span multiple floppy disks. What I particularly like about PKZIP is that it can update an existing ZIP file with new and modified files, meaning that the ZIP file will always contain current versions of all files. Unzip the files and everything is restored. With MSBACKUP, you need to restore the full backup, then the differential backup or multiple incremental backups, to fully restore all files.

Although my PC has a tape drive, I do routine PKZIP "backups" through the batch files that run my applications every day. Because I have two hard drives (I keep most of my data on drive D), I actually do two routine backups. First I ZIP the files in a subdirectory into a file on drive C; then I copy that ZIP file to floppies. This strategy gives me copies of my important data files on both hard disks, and on floppies in case both drives are inaccessible.

Each application I run has its own batch file, with the same type of backup logic. ARTICLES.BAT (see the program listing, page 26) is the batch file I use for my writing. As

you can see, this batch file first takes me into WordPerfect, my word processor. After exiting WP, the batch file asks whether I want to back up the \ARTICLES directories containing my WP documents. If the answer is yes, it then ZIPs the

A SWITCH IN TIME

MSBACKUP has been criticized for forcing you to go through several layers of menus before you can start a backup. An undocumented /BATCH switch for MSBACKUP, however, can automate most of a routine backup.

Start MSBACKUP, select BACKUP, then select all settings for the specific backup you want to automate. Select the files to back up, the backup type (full, incremental, differential), and the Backup To drive. Under Options, check the Quit After Backup box and uncheck the Prompt Before Overwriting Used Diskettes box. Save all these settings in a new setup file (File, Save Setup As)—for example, MYDATA.SET.

Now you can run MSBACKUP automatically by giving it the setup filename and including the /BATCH switch:

```
MSBACKUP MYDATA /BATCH
```

MSBACKUP starts the backup and prompts you to insert each floppy disk. Insert the disk and press Enter or click on the Continue button. When the backup is completed, MSBACKUP will close.

As as alternative, you can use the XCOPY command as part of your incremental backup strategy because it can easily copy only modified files. It does this through the /M switch, which copies files with the archive attribute turned on, then turns the archive attribute off after the copy is successful.

Using XCOPY as a backup technique is best done in a batch file. Because XCOPY can't format disks like MSBACKUP or BACKUP, you'll need to start with blank, formatted disks. Start XCOPY with the /M switch. When a disk fills, XCOPY will end with an ERRORLEVEL of 4. The batch file can check for this with an IF statement and loop back to the XCOPY command. Previously copied files won't be copied again because their archive attributes will be off.

Here's a sample batch file that does an incremental backup of all modified files in the C:\DATA directory, including child subdirectories and their files:

```
@ECHO OFF
:LOOP
ECHO Insert blank, formatted disk in A: drive
PAUSE
XCOPY C:\DATA\*.* A: /S /M
IF ERRORLEVEL 4 GOTO LOOP
ECHO *** All Done ***
```

XCOPY doesn't copy hidden or system files—those with their hidden or system attributes turned on. (The DOS 5 *Users Guide* says that in previous DOS versions XCOPY copied hidden and system files, but that's incorrect.) You typically should avoid copying or moving system files, which often contain absolute disk addresses and must be in specific physical locations. Examples of such system files are IO.SYS, MSDOS.SYS, and the Windows permanent swap file 386SPART.PAR. And, of course, XCOPY can't copy a file that's larger than the disk to which you're copying.

—K.J.

files to the C:\BACKUPS directory on the other hard drive. The PKZIP switches update (-U) the existing ZIP files with new and modified files, and use extra compression (-EX) to keep the ZIP files as small as possible.

After zipping all the data directories, the batch file asks whether to copy those ZIP files to floppy disk. If the answer is yes, it uses the REPLACE command with the /UPDATE (/U) switch to copy only ZIP files more recent than the files already on the floppy. (If the ZIP file isn't newer, none of its files has been updated, so there's no reason to copy it.) Because these ZIP files are large, they're copied to three numbered disks. (See "Backing Up Compressed Drives," right, for information on using DoubleSpace and DriveSpace.)

Getting into a Routine

Protecting your data doesn't have to be a chore. Backups can be easier and less time-consuming if you know which files to back up, if you run a backup cycle with a full backup and then an incremental or a differential backup, and if you automate the process with working batch files. One of these days you'll be glad you did. ∎

BACKING UP COMPRESSED DRIVES

If you're using DoubleSpace or DriveSpace, it's important to understand which files to back up. You want to back up the files on the compressed drive, *not* the compressed volume file (CVF) on the host (uncompressed) drive. The CVF has a name such as DBLSPACE.000 (MS-DOS 6.20) or DRVSPACE.000 (MS-DOS 6.22), with read-only, hidden, and system attributes.

For example, if you've compressed drive C and DoubleSpace has created the host drive as drive D, you should back up all your data files on drive C. You don't need to back up the DBLSPACE.000 file on drive D (although you should back up any other data files on drive D). Backing up the files from drive C lets you restore individual files from the backup. If you back up only the DBLSPACE.000 file, you can restore only the entire drive. That can be a problem, because all changes to files on the compressed drive since the backup will be lost when the DBLSPACE.000 file is restored.

There are a couple of other reasons for backing up the compressed-drive files and not the host CVF. When you back up individual files, you can restore them on any PC—not just one running DoubleSpace or DriveSpace. If you back up only the CVF, you must restore it on a computer with DoubleSpace or DriveSpace; if not, the data on the CVF will be inaccessible. Moreover, if you do back up the compressed drive files, backing up the CVF will also add significant time and space requirements with no real benefit. If you have to restore files, you'll use the compressed-drive backup, not the backed-up CVF file.

—K.J.

ARTICLES.BAT takes the author into WordPerfect, and after exiting asks whether he wants to back up the directory containing the documents. It then ZIPs the files to a directory on another hard drive. Note that any indented line beginning with D: in the left-hand column is actually overflow from the line above it (belongs on the same line when you type it in on your screen).

```
@ECHO OFF
WP /R /W=*,* /M=PICKDIR
ECHO ****************************************
ECHO *    Back up ARTICLES directories ?    *
ECHO *    Press "Y" to back up or "N" to ignore *
ECHO ****************************************
CHOICE
IF ERRORLEVEL 2 GOTO END


ECHO.
ECHO Creating ZIP files....
ECHO.
PKZIP C:\BACKUPS\ARTICLES.ZIP -U -EX D:\ARTICLES\*.*
PKZIP C:\BACKUPS\TIPS.ZIP -U -EX
   D:\ARTICLES\TIPS\*.*
PKZIP C:\BACKUPS\PUBLISH.ZIP -U -EX
   D:\ARTICLES\PUBLISH\*.*
PKZIP C:\BACKUPS\LAWOFF.ZIP -U -EX
   D:\ARTICLES\LAWOFF\*.*
PKZIP C:\BACKUPS\DOSWORLD.ZIP -U -EX
   D:\ARTICLES\DOSWORLD\*.*
PKZIP C:\BACKUPS\OTHER.ZIP -U -EX
   D:\ARTICLES\OTHER\*.*
PKZIP C:\BACKUPS\CCS.ZIP -U -EX D:\ARTICLES\CCS\*.*
PKZIP C:\BACKUPS\GRAPHICS.ZIP -U -EX
   D:\ARTICLES\GRAPHICS\*.*

ECHO.
```

```
ECHO ****************************************
ECHO *    Back up to Floppy Disks? If Yes, insert  *
ECHO *     backup disk NUMBER 1 in B: drive.       *
ECHO ****************************************
ECHO.
CHOICE "BACK UP TO THE B: DRIVE "
IF ERRORLEVEL 2 GOTO END

REPLACE C:\BACKUPS\ARTICLES.ZIP B: /U
REPLACE C:\BACKUPS\PUBLISH.ZIP B: /U
REPLACE C:\BACKUPS\CCS.ZIP B: /U
REPLACE C:\BACKUPS\LAWOFF.ZIP B: /U

ECHO.
ECHO Insert backup disk NUMBER 2 in B:
ECHO.
PAUSE
REPLACE C:\BACKUPS\DOSWORLD.ZIP B: /U
REPLACE C:\BACKUPS\OTHER.ZIP B: /U
REPLACE C:\BACKUPS\TIPS.ZIP B: /U


ECHO.
ECHO Insert backup disk NUMBER 3 (GRAPHICS) in B:
ECHO.
PAUSE
REPLACE C:\BACKUPS\GRAPHICS.ZIP B: /U

:END
ECHO *** All Done ***
```

End

PC TOOLBOX

Exploring Inner Space

Will it fit or won't it? Copying or moving files doesn't have to be a gamble. You'll know ahead of time whether you have the space you need by incorporating these two Debug utilities into your batch programs.

by Robert L. Hummel

You'll never know until you try." Those words of wisdom have served me well in many situations: staring at a plate of calamari or standing in an open hatchway at 12,000 feet with a parachute strapped to my back, for example. But, when it comes to computers, blind faith is often a recipe for trouble. When your data is on the line, the wiser advice may be "Look before you leap."

Bailing Out

Certain DOS operations have always bothered the overly cautious side of me. The COPY command, for instance, always seems to have me leaping when I should be looking.

For example, COPY used to have a distressing habit of overwriting files of the same name without any

Contributing Editor Robert L. Hummel is the author of PC Magazine Programmer's Technical Reference: The Processor and Coprocessor, PC Magazine Assembly Language Lab Notes, *and* Programmer's Technical Reference: Data and Fax Communications *(all from Ziff-Davis Press, 1992).*

warning. DOS 6.2 addressed this problem, and now COPY prompts you before destroying files.

But another problem with COPY, XCOPY, MOVE, and similar DOS commands has been overlooked. In all cases, these types of DOS commands neglect to determine whether there's enough room on the destination disk for the files they plan to put there.

Watching as a long list of files is transferred, only to have it abort uncompleted because the disk is full, is annoying. But in a batch program it can be fatal.

The COPY Problem

Operation of the COPY command is straightforward: It attempts to copy files one at a time from a source to a destination.

But because COPY is an internal DOS command, it doesn't return an exit code; there's no way for your batch program to know whether the copy succeeded or failed.

This can often lead to compounded mistakes and disaster. For example, consider the following two-line batch program:

```
COPY *.* A:
DEL *.*
```

Obviously, the intent here is to copy all files from the current directory to a floppy, after which the original files are deleted. The program works fine—until you try to copy 500K of files to a floppy that has less than 500K of free space. In that case, COPY simply copies files until they no longer fit, then aborts the operation.

The problem is that there's no way for the batch program to know this. So in this case, the program will blithely delete the original files—even those that haven't been preserved on the floppy.

The WillFit Solution

You can't protect yourself against failures like this after the fact. Instead, the best approach is to prevent the failure from happening at all. The time to act is before you use the COPY command. Because DOS doesn't provide this capability, I wrote the utility WILLFIT.COM.

Before you can use WILLFIT, you'll have to create it. To convert the

WILLFIT.SCR creates the WILLFIT.COM program, which determines whether a file or a group of files will fit on a target disk.

```
N WILLFIT.COM                    MOV    AH,4F
A 100                            INT    21
CLD                              JB     14F
SUB    BX,BX                     ADD    BP,[9A]
OR     AL,AL                     ADC    DI,[9C]
JNZ    10C                       JMP    13F
INC    BX                        MOV    DL,[6C]
OR     AH,AH                     OR     DL,DL
JZ     112                       JZ     10C
MOV    AL,BL                     INC    BX
MOV    AH,4C                     MOV    SI,BX
INT    21                        MOV    AH,36
INC    BX                        INT    21
MOV    SI,81                     DB     93
LODSB                            MUL    CX
CMP    AL,D                      JMP    167
JZ     10C                       SHL    AX,1
CMP    AL,20                     RCL    DX,1
JZ     116                       SHR    BX,1
DEC    SI                        OR     BX,BX
MOV    DX,SI                     JNZ    163
INC    BX                        MOV    BX,SI
LODSB                            CMP    DI,DX
CMP    AL,D                      JB     179
JZ     10C                       JA     10C
CMP    AL,20                     CMP    BP,AX
JNZ    123                       JA     10C
MOV    Byte Ptr [SI-1],0         MOV    BL,FF
SUB    DI,DI                     JMP    10C
SUB    BP,BP
MOV    AH,4E                     RCX
MOV    CX,1                      7D
INT    21                        W
JB     14F                       Q
JMP    145                                        End
```

This batch program, COPIER, uses WILLFIT to determine whether its files will fit on the target drive.

```
WILLFIT *.* A:

IF ERRORLEVEL 255 GOTO OKAY

SET ERROR=Source drive invalid
IF ERRORLEVEL 1 SET ERROR=Target drive invalid
IF ERRORLEVEL 2 SET ERROR=Source missing
IF ERRORLEVEL 3 SET ERROR=Target missing
IF ERRORLEVEL 4 SET ERROR=Files won't fit
ECHO %ERROR%
GOTO END

:OKAY
COPY *.* A:
DEL *.*

:END                                              End
```

Debug script WILLFIT.SCR, shown in the first listing (top left), to the executable program WILLFIT.COM, follow the instructions in the "Debug Scripts" section of "How to Use This Magazine" (page 63 in this issue).

The syntax for WILLFIT is basically the same as the COPY command's:

```
WILLFIT [path]filename.ext d:
```

The first argument is the source-file specifier, which can include a drive and path and wild cards. The second argument is the drive to which you intend to copy these files. WILLFIT locates all files that match the source specification, totals their sizes, and compares the result with the free space that's reported on the target drive.

WILLFIT performs extensive error checking and—unlike COPY—returns an exit code you can test within your batch programs. The accompanying table (opposite) lists and defines the exit codes WILLFIT returns.

Check It Out

So let's take the previous batch example and rewrite it using WILLFIT. The result is shown in the second listing (bottom left).

Now we've covered all the bases. Unless WILLFIT returns an exit code of 255, the program won't even attempt to copy the files. Instead, it analyzes the exit code and reports the error to the user as an informative text message.

Under DOS, if files on the destination drive have the same names as the source files, COPY overwrites the destination files. By doing so, the space taken up by the overwritten files is effectively freed. But WILLFIT just checks for free space, not space that will be freed by a subsequent copy. As a result, WILLFIT may give a false "no room" message in some circumstances. But WILLFIT never gives a false positive.

Using WILLFIT is appropriate before copying from a hard disk to

a floppy—or vice versa. Of course, it's unlikely that the files on a single floppy will exceed the free space on a hard disk.

But what if you're copying files from two, or ten, or two dozen floppies? Or even between two hard drives? Because WILLFIT can't determine the sizes of files in subdirectories or across disks, a new utility is called for.

DRVFREE Tells All

At the command line, you can execute a DIR or CHKDSK command to

WILLFIT.COM returns the exit codes shown to report the result of an intended copy operation.

Type	Description
0	source drive invalid
1	destination drive invalid
2	source-file specification missing
3	destination-drive specification missing
4	files won't fit on destination drive
255	files will fit on destination drive

see how much free space a disk contains. But batch programs aren't as versatile—they can't read the screen conveniently.

In the end, it was simpler to write the utility DRVFREE.COM. Create it from the Debug script DRVFREE.SCR, shown in the third listing (right).

DRVFREE's syntax is simple:

```
DRVFREE [d:]
```

where *d:* is the drive you want to examine. DRVFREE calculates the number of free bytes on the specified drive, truncates to the nearest million bytes, then returns that number as an exit code. If no drive is specified, DRVFREE examines the current drive.

Because an exit code must fall in the range 0 (zero) to 255, DRVFREE's precision is limited. A drive with 999,999 bytes free, for example,

will return 0 (zero). And all drives with 254,000,000 or more bytes free return 254.

DRVFREE reserves the value 255 to report an error condition, such as an invalid drive.

Putting DRVFREE to Work

The only thing worse than a full hard disk is one that fills up when you're not expecting it. To avoid nasty surprises, I've added the following lines to the end of my AUTOEXEC.BAT file:

```
DRVFREE C:
IF ERRORLEVEL 20
  GOTO END
ECHO Warning!
ECHO Drive C: has
ECHO less than
ECHO 20,000,000
ECHO bytes free!
PAUSE
:END
```

Now each time I boot up, the DRVFREE utility checks my disk to ensure that I've got at least 20 million bytes free. If not, my AUTOEXEC.BAT file displays a warning message and pauses. I can then either hit a key to continue or stop

for a few moments to take some remedial action, such as deleting or backing up excess files.

Similarly, here's a batch program that restores files from a ZIP file archive that spans several disks. In this case, DRVFREE returns the free space on drive C. If less than 8,000,000 bytes' worth is free, the batch program won't unpack the files:

```
DRVFREE C:
IF ERRORLEVEL 8 GOTO UNPACK
ECHO This ZIP file requires
ECHO at least 8,000,000 bytes
ECHO when unpacked.
GOTO END
:UNPACK
PKUNZIP DOOM.ZIP C:
:END
```

Spatial Relations

DOS batch-program operations shouldn't be games of chance. And, with the WILLFIT and DRVFREE utilities presented here, they don't have to be.

Protect yourself against surprises by using WILLFIT to ensure that your files will fit before you copy them, and DRVFREE to monitor your hard-drive space. With these two utilities, you can be safe—instead of sorry. ■

DRVFREE.SCR creates the DRVFREE.COM program, which returns a drive's free space as an exit code.

```
N DRVFREE.COM          MOV   SI,AX
A 100                  MOV   AX,DX
OR    AL,AL            SUB   DX,DX
JZ    10A              DIV   BX
MOV   AL,FF            MOV   DI,AX
MOV   AH,4C            MOV   AX,SI
INT   21               DIV   BX
MOV   AH,36            MOV   DX,DI
MOV   DL,[5C]          LOOP  127
INT   21               CMP   AL,FE
DB    93               JBE   106
MUL   CX               MOV   AL,FE
JMP   11B              JMP   106
SHL   AX,1
RCL   DX,1
SHR   BX,1             RCX
OR    BX,BX            41
JNZ   117              W
MOV   BX,3E8           Q
MOV   CX,2
                                    End
```

Applied Math: Simple as 1, 2, 3

Adding and subtracting with a batch-file calculator can help you create your own file-maintenance routines.

by Hardin Brothers

A couple of issues back, I explained how to use the undocumented slash character in batch files and credited it with all sorts of miraculous powers. Some of you wrote disbelieving letters, particularly about my claim that batch programs could perform arithmetic.

The accepted wisdom is that batch files can't do math. But, as is often the case in the computer world, the accepted wisdom isn't quite right, as we'll see with this month's calculator program.

CALC.BAT (page 32) can add and subtract numbers of up to four digits. Most of the program is simple, but I use a couple of advanced techniques. If you weren't around for my discussion of the slash in the FOR... IN...DO command or how to call subroutines in a batch file ("A Batch-File Breakthrough," parts 1 and 2, *DOS World* #19, January 1995, page 36, and #20, March 1995, page 36, respectively), take a look at my columns covering those topics.

Technical Editor Hardin Brothers has been working with computers and writing about them for 15 years.

Mathing Around

To use CALC.BAT, type CALC followed by ADD or SUB and then by the two numbers to add or subtract.

For example, if you want to add 728 and 493, type this line:

```
CALC ADD 728 493
```

CALC.BAT assumes that the first parameter is a command—in this case, ADD. To work with subroutines, CALC often calls itself with a command. For example, partway through processing the addition you just typed in, CALC restarts itself with this command:

```
CALC ADD_1D 8 3 AC1 CARRY
```

As it restarts, CALC recognizes ADD_1D as a command.

To sort out the commands, CALC starts by storing its own name (in line 7) and then jumping to the required command in line 9. To keep the program from growing voluminously, there's no error checking at the beginning. If you start the program without a recognized command, it simply issues an error message and ends.

The entire program after the first nine lines is made up of small routines. The words you type when you boot CALC start the ADD and SUB routines in lines 11 to 31. The program itself calls the remainder of the routines.

ADD and SUB are pretty simple. Because the program is relatively slow—you may be able to do the arithmetic in your head faster than the batch file can calculate an answer—both ADD and SUB start by ECHOing "Thinking . . ." to the screen to show that something is going on. And then each of those two modules calls another subroutine to do the real work.

Addition and Subtraction

The remainder of the program is easier to understand if we start with the simplest and lowest-level subroutines: those that do the actual calculations.

INC and DEC (lines 36 to 67) have the job of adding or subtracting 1 from a digit and returning the result in an environment variable. All arithmetic is based on counting up and down; these are the routines that do the counting.

The next two routines, ADD_1D and SUB_1D (lines 69 to 102 and lines 104 to 137), are responsible for adding and subtracting pairs of digits. In the sample problem above, for example, ADD_1D adds 8 + 3, 2 + 9, and 7 + 4. ADD_1D (add one digit) and SUB_1D are almost identical, except that one is concerned with carrying and the other with borrowing.

To add two digits, ADD_1D calls INC to increment one digit and DEC to decrement the other. When the second digit contains zero, the addition is finished.

For example, to add 3 and 2, SUB_1D calls INC and DEC to change the digits to 4 and 1, and then INC and DEC again to change them to 5 and zero. Because the second number now has zero, the first digit has the result.

If, at any time, the first digit changes to zero, a carry into the next position has occurred and ADD_1D sets the carry flag (contained in %4) to 1.

After the routine has finished adding, it checks to see whether there's a carry into the current column. If so, it calls INC one more time and then checks once again for a carry.

Subtraction, as implemented by SUB_1D, works in a similar way, except that DEC operates on both digits and INC isn't called at all. To subtract two digits, the program decrements them both. If, at any time, the first digit changes to a 9, a borrow from the next column has occurred. When the second digit contains zero, the first digit contains the results of the subtraction.

For example, to subtract 2 from 5, the digits are 5 and 2, 4 and 1, and 3 and zero. Since the second digit is zero, the first digit contains the result, which is 3.

The ADD_1D and SUB_1D routines are the heart of the program and

the most difficult to understand. The rest of the program simply manipulates the numbers to switch between the format the program requires and the format humans expect.

Digital Manipulations

The DIGITS routine (lines 139 to 168) is simple. It receives a single value of four digits or fewer. (Longer numbers are simply truncated after four digits.) It breaks the number into four individual digits and returns them in four environment variables. But what happens when the original number contains fewer than four digits? The DIGITS routine handles that possibility by padding the number with zeros on the left until it's four digits wide.

DIGITS begins calling a routine named REVERSE to switch the number it receives. For example, if DIGITS receives the number 123, REVERSE turns the number around to 321. Next, DIGITS adds four zeros to the end of the number. In our example, the result is 3210000.

Then DIGITS uses the special FOR...IN...DO slash to extract the first four characters in that string. It extracts 3 as the ones digit, 2 as the tens digit, 1 as the hundreds digit, and zero as the thousands digit. It discards the remainder of the number (in this case 000).

The REVERSE routine (lines 170 to 188) returns in reverse order any string it receives. In this program, it reverses the number so that the ones digit comes first. Like DIGITS, the REVERSE routine uses the FOR...IN...DO slash to break a string into individual characters.

As it removes each character or digit from the front of the string it has received, it puts that character or digit at the front of its result string. That may sound backwards at first, but watch the way the

string reverses. The first column below shows the original string; the second column holds the result after each step:

```
1234
234     1
34      21
4       321
        4321
```

Routine Coordination

All the pieces are now in place. The last two routines, ADD4 and SUB4 (lines 190 to 234), are the real workhorses of the program. Each receives the two values you typed when you started CALC. They pass each number to DIGITS to separate individual digits, then send pairs of digits to ADD_1D or SUB_1D for processing. Finally, they put it all back together to get a single result.

Besides the way they handle carrying or borrowing from one column to another, these two routines have another important difference. When they're finished, the ADD4 routine checks to see whether there's a carry into the fifth (ten thousands) column. If so, it adds a 1 to the beginning of the answer and returns a five-digit result.

The SUB4 program can't work that way. If a borrow remains at the end, that means you tried to subtract a larger number from a smaller one. This calculator isn't set up to handle negative numbers, so SUB4 sets the result to Error instead of a number.

Neat Tricks

There's one other caveat. If the second number includes a character that's not a digit, the program will enter an endless loop, because that character will never turn to zero to signal the end of an ADD_1D or SUB_1D sequence.

If the first digit contains a non-numeral character, the answer will

CALC.BAT adds and subtracts numbers containing up to four digits; in your own batch files it can assist in file-maintenance chores such as date comparison, determining available disk space before copying, and so on. Line numbers are for reference only; don't type them in.

```
 1 @ECHO OFF
 2 :: CALC.BAT calculates the sum or difference
 3 :: of 2 numbers that may be up to 4 digits long each.
 4
 5 :: Save this program's name and branch to the
 6 :: correct subroutine:
 7 SET My=%0
 8 Shift
 9 GOTO %0
1Ø
11 :ADD
12 ::   To add 2 numbers, start this
13 ::    program as Calc Add x1 x2
14 ECHO.
15 ECHO Thinking...
16 CALL %My% ADD4 %1 %2 Result
17 ECHO %1 + %2 = %Result%
18 ECHO.
19 SET Result=
2Ø GOTO End
21
22 :SUB
23 ::   To subtract 2 numbers, start this
24 ::    program as Calc Sub x1 x2
25 ECHO.
26 ECHO Thinking...
27 CALL %My% SUB4 %1 %2 Result
28 ECHO %1 - %2 = %Result%
29 ECHO.
3Ø SET Result=
31 GOTO End
32
33 :: Below are the subroutines needed for
34 :: addition and subtraction
35
36 :INC
37 ::   INC gets 2 parameters: a digit as %1
38 ::   and the name of an env. variable as %2
39 ::   Result:  Set %2=%1 + 1
4Ø ::   No error checking or carry checking here.
41 IF %1!==Ø! SET %2=1
42 IF %1!==1! SET %2=2
43 IF %1!==2! SET %2=3
44 IF %1!==3! SET %2=4
45 IF %1!==4! SET %2=5
46 IF %1!==5! SET %2=6
47 IF %1!==6! SET %2=7
48 IF %1!==7! SET %2=8
49 IF %1!==8! SET %2=9
5Ø IF %1!==9! SET %2=Ø
51 GOTO End
52
53 :DEC
54 ::   DEC gets 2 parameters: a digit as %1
55 ::   and the name of an env. variable as %2
56 ::   Result: Set %2=%1 - 1
57 IF %1!==Ø! SET %2=9
58 IF %1!==1! SET %2=Ø
59 IF %1!==2! SET %2=1
6Ø IF %1!==3! SET %2=2
```

```
61 IF %1!==4! SET %2=3
62 IF %1!==5! SET %2=4
63 IF %1!==6! SET %2=5
64 IF %1!==7! SET %2=6
65 IF %1!==8! SET %2=7
66 IF %1!==9! SET %2=8
67 GOTO End
68
69 :Add_1D %1 %2 %3 %4 %5
7Ø :: Add_1D gets 4 or 5 parameters: %1 and %2 are
71 :: digits.  %3 is the env. variable that will
72 :: hold the result. %4 is another env. variable
73 :: that will hold the carry status (Ø or 1)
74 :: The 5th parameter, if it exists, must be
75 :: a "Ø" or "1" and is the carry out of the
76 :: last addition)
77 ::     Result:  Set %3= %1 + %2 + %5
78 ::              if carry, Set %4=1 else Set %4=Ø
79 SET %4=Ø
8Ø SET a1=%1
81 SET a2=%2
82
83 :Add_1D_1
84 ::   This is the main addition loop
85 IF %a2%!==Ø! GOTO Add_1D_2
86 CALL %My% INC %a1% a1
87 CALL %My% DEC %a2% a2
88 IF %a1%!==Ø! SET %4=1
89 GOTO Add_1D_1
9Ø
91 :Add_1D_2
92 ::   End of the loop.  Add in carry (%5)
93 IF NOT '%5'=='1' GOTO Add_1D_3
94 CALL %My% INC %a1% a1
95 IF '%a1%'=='Ø' SET %4=1
96
97 :Add_1D_3
98 ::   Clean up and leave
99 SET %3=%a1%
1ØØ SET a1=
1Ø1 SET a2=
1Ø2 GOTO End
1Ø3
1Ø4 :Sub_1D
1Ø5 :: Sub_1D gets 4 or 5 parameters: %1 and %2 are
1Ø6 :: digits.  %3 is the env. variable that will
1Ø7 :: hold the result. %4 is another env. variable
1Ø8 :: that will hold the carry status (Ø or 1)
1Ø9 :: The 5th parameter, if it exists, must be
11Ø :: a "Ø" or "1" and is the borrow of the
111 :: last subtraction)
112 ::     Result:  Set %3= %1 - %2 - %5
113 ::              if borrow, Set %4=1 else Set %4=Ø
114 SET %4=Ø
115 SET s1=%1
116 SET s2=%2
117
118 :Sub_1D_1
119 ::   This is the main subtraction loop
12Ø IF '%s2%'=='Ø' GOTO Sub_1D_2
```

```
121 CALL %My% DEC %s1% s1
122 CALL %My% DEC %s2% s2
123 IF '%s1%'=='9' SET %4=1
124 GOTO Sub_1D_1
125
126 :Sub_1D_2
127 ::  End of the loop.  Subtract borrow (%5)
128 IF NOT '%5'=='1' GOTO Sub_1D_3
129 CALL %My% DEC %s1% s1
130 IF '%s1%'=='9' SET %4=1
131
132 :Sub_1D_3
133 ::  Clean up and leave
134 SET %3=%s1%
135 SET s1=
136 SET s2=
137 GOTO End
138
139 :Digits
140 ::  Receives a numeric value in %1
141 ::  and the names of 4 variables in %2 %3 %4 %5
142 ::  Puts the 1's digit in %2, the 10's digit in %3
143 ::  the 100's digit in %4, and the 1000's digit in %5
144 ::  If %1 is shorter than 4 digits, pads on the left
145 ::  with 0's
146
147 CALL %My% Reverse %1 d1
148 SET d1=%d1%0000
149 SET d2=%d1%
150
151 FOR %%c IN (/%d2%) DO SET d1=%%c
152 FOR %%c IN (/%d2%) DO IF %%c%d1%==%d2% SET %2=%%c
153 SET d2=%d1%
154
155 FOR %%c IN (/%d2%) DO SET d1=%%c
156 FOR %%c IN (/%d2%) DO IF %%c%d1%==%d2% SET %3=%%c
157 SET d2=%d1%
158
159 FOR %%c IN (/%d2%) DO SET d1=%%c
160 FOR %%c IN (/%d2%) DO IF %%c%d1%==%d2% SET %4=%%c
161 SET d2=%d1%
162
163 FOR %%c IN (/%d2%) DO SET d1=%%c
164 FOR %%c IN (/%d2%) DO IF %%c%d1%==%d2% SET %5=%%c
165
166 SET d1=
167 SET d2=
168 GOTO End
169
170 :Reverse
171 ::  Gets a string in %1 and the name of an
172 ::  env. variable in %2.  Reverses %1 and places
173 ::  the result in %2
174
175 SET Rv1=%1
176 SET Rv4=
177
178 :Rev_1
179 SET Rv2=%Rv1%
180 SET Rv3=%Rv1%
181 FOR %%c IN (/%Rv2%) DO SET Rv1=%%c
182 FOR %%c IN (/%Rv2%) DO IF %%c%Rv1%==%Rv2% SET Rv3=%%c
183 SET Rv4=%Rv3%%Rv4%
```

```
184 IF NOT %Rv2%==%Rv1% GOTO Rev_1
185
186 SET %2=%Rv4%
187 FOR %%v IN (Rv1 Rv2 Rv3 Rv4) DO SET %%v=
188 GOTO End
189
190 :ADD4
191 ::  Receives 2 values in %1 and %2 and
192 ::  the name of an env. variable in %3
193 ::  Results: Set %3= %1 + %2
194
195 SET %3=
196
197 CALL %My% digits %1 aa1 aa2 aa3 aa4
198 CALL %My% digits %2 ab1 ab2 ab3 ab4
199
200 CALL %My% Add_1d %aa1% %ab1% ac1 carry
201 CALL %My% Add_1d %aa2% %ab2% ac2 carry %carry%
202 CALL %My% Add_1d %aa3% %ab3% ac3 carry %carry%
203 CALL %My% Add_1d %aa4% %ab4% ac4 carry %carry%
204
205 IF NOT %carry%!==1! SET %3=%ac4%%ac3%%ac2%%ac1%
206 IF %carry%!==1!     SET %3=1%ac4%%ac3%%ac2%%ac1%
207
208 FOR %%v IN (aa1 ab1 ac1 aa2 ab2 ac2) DO SET %%v=
209 FOR %%v IN (aa3 ab3 ac3 aa4 ab4 ac4) DO SET %%v=
210 SET carry=
211 GOTO End
212
213 :SUB4
214 ::  Receives 2 values in %1 and %2 and
215 ::  the name of an env. variable in %3
216 ::  Results: Set %3= %1 - %2
217 ::  If %2 > %1, Set %3=Error!
218
219 SET %3=
220
221 CALL %My% digits %1 sa1 sa2 sa3 sa4
222 CALL %My% digits %2 sb1 sb2 sb3 sb4
223
224 CALL %My% Sub_1d %sa1% %sb1% sc1 borrow
225 CALL %My% Sub_1d %sa2% %sb2% sc2 borrow %borrow%
226 CALL %My% Sub_1d %sa3% %sb3% sc3 borrow %borrow%
227 CALL %My% Sub_1d %sa4% %sb4% sc4 borrow %borrow%
228
229 SET %3=%sc4%%sc3%%sc2%%sc1%
230 IF %borrow%!==1! SET %3=Error
231
232 FOR %%v IN (sa1 sb1 sc1 sa2 sb2 sc2) DO SET %%v=
233 FOR %%v IN (sa3 sb3 sc3 sa4 sb4 sc4) DO SET %%v=
234 SET borrow=
235
236 :End
```

End

be nonsensical. A finished and polished batch file would never let such an error occur.

The best place to check for non-numeral entries would probably be in REVERSE, which extracts each digit individually. But that would change the logic of REVERSE so that it would work only with numbers.

If you want to add such a check, put the following four lines into REVERSE between lines 182 and 183 (note that the second line shows overflow here, but that the DO section should be part of the FOR line when you're typing this in):

```
SET RvEr=1
FOR %%N IN (Ø 1 2 3 4 5 6 7 8 9)
   DO IF %%N!==%Rv3%! SET RvEr=Ø
IF RvEr!==1! ECHO Error: Bad Entry
IF RvEr!==1! ECHO Ctrl+C | Pause > Nul
```

Also, in the last line above, replace Ctrl+C with ASCII character 3. If you're using EDIT.COM, press Ctrl+P and then Ctrl+C; you'll see a heart-shaped character on screen representing Ctrl+C.

These lines work by first setting the environment variable RvEr to 1 as an error flag. The second line checks to make sure the character in Rv3 is actually a digit. If it is, the error flag is set to zero. The last two lines check the error flag. If it's still 1, an error has occurred. First, an error message is displayed on screen; then the program ends.

That last line is a neat way to end in the middle of a batch file and return to the DOS prompt. It works in a main batch file, in a batch file that has been started by another, in a batch file that has been called by another, and in a batch file that has called itself. It's a way of telling COMMAND.COM, "Stop all batch-file processing and stop it right now!" The user has to verify that batch processing should stop, but that's a simple matter of pressing Y or N and should be automatic after the error message.

Final Thoughts

CALC.BAT proves that batch files can perform arithmetic. You can use the program's techniques in your own batch files to determine whether disk space is available to copy files, to compare dates of files, or to perform similar tasks.

Because CALC.BAT isn't fast and can add or subtract only integers, you'll probably want to look elsewhere for a true command-line calculator. If you use JP Software's 4DOS or Symantec's NDOS (in Norton Utilities), you can create a better calculator with this command:

```
Alias Calc=`ECHO Answer is: %@veal[%&]`
```

That's a lot less work than entering CALC.BAT by hand, and it produces a much more powerful calculator. You can also find fast and capable command-line calculators on BBSes and on-line services.

The strength of its competition, however, doesn't limit CALC.BAT's uses. Once you understand the techniques, you're ready to create more dynamic batch files of your own. ■

DOS TIP — ONE THING AFTER ANOTHER

You may not know it, but DOS sometimes lets you execute several commands in a single line. To prove it, type this command:

```
CD directory | DIR
```

where *directory* is the name of a subdirectory one level below the current directory. When you press the Enter key, DOS will do two things: change to the specified directory and then display a directory of this subdirectory. This technique works because of the pipe (|), a symbol which tells DOS to redirect the output from one command to a temporary file. DOS then sends the information in that file as input to another command.

You may use this piping trick with other commands, as well. For example, to delete two files called ONE and TWO, you may type:

```
DEL ONE | DEL TWO
```

To define several DOSKEY macros, you might type:

```
DOSKEY D = DIR | DOSKEY T = TREE
```

This technique also works for renaming several files or making or removing several directories. Because error messages aren't redirected, they appear on the screen as usual.

However, because the pipe symbol isn't meant to be used as a command separator, the technique produces unexpected results, unless you know exactly how each command in the chain works. For example, you might expect this command to display two directory listings:

```
DIR ONE | DIR TWO
```

That's not the case; you see only the directory of \TWO. This happens because the output of the first command, DIR ONE, is sent to a temporary file rather than being displayed. Because the second command is incapable of displaying input from the pipe, DOS never sends it, and therefore you never see it.

The moral? Exercise caution when attempting to execute multiple commands. If you're not certain how a particular command handles output or input, don't include it in a command that uses this piping trick.

—Richard Penn

Symbolic Victory: Part 1

To create even the simplest batch files, you need to master DOS's assortment of symbols. Here's the scoop on six of the 14 indicators you should know.

by Eric Maloney

Writing batch files without symbols is like balancing your budget without plus, minus, and equal signs. Even the most rudimentary batch files include at least one symbol, the at sign (@), which keeps DOS from displaying a batch file's command.

Fortunately, there aren't many symbols to learn—14 in all, fewer if you consider that some are variations of others. And although some of them are confusing at first (what's the difference between = and ==? or % and %%?), most aren't inherently complicated. Even if you're a batch-file beginner, you'll need only a little time and patience to figure out the basics.

The easiest way to explain the symbols is to discuss them one by one and provide a few simple examples showing them in action. In addition, I've included a more

Editor at Large Eric Maloney is the founding editor of PC Resource *and the former editor in chief of* 80 Micro *and* LAN Technology *magazines. He's currently adjunct professor of journalism at the University of New Hampshire/Keene.*

elaborate batch file, called PHONE-.BAT (see the listing, page 36), which provides further examples. (See the sidebar, "Using PHONE.BAT," page 38, for details.) PHONE.BAT uses all the symbols discussed here, except the slash (/), which I'll cover in the next installment (*DOS World* #23, September 1995). The accompanying table, "Symbol Summary" (page 39), outlines each item's function and lists the line in PHONE.BAT in which it first appears.

The At Sign (@)

Placed at the beginning of a line, an at sign (@) prevents the remainder of the commands and statements in that line from being displayed. For example, in the following line:

```
@ECHO This is a test.
```

DOS displays the words "This is a test." but not the command ECHO. To use the at sign, you must have DOS 3.3 or later.

The Period (.)

When it follows an ECHO statement, as it does in line 2 of PHONE.BAT, a

period sends a carriage return to the screen. This produces a blank line, which gives text on screen a neater appearance. (For examples, see the various help and message screens starting at line 103 of PHONE.BAT.)

Note that if you ECHO a line of periods to the screen (to set off text, for example), the first one isn't displayed. For instance, the following command produces just three periods:

```
ECHO....
```

To produce four periods, you must type an extra period or put a space between ECHO and the first period:

```
ECHO ....
```

The Percent Sign (%)

A single percent sign indicates a variable to which you assign a value from the DOS prompt. You must follow it with a single digit from 0 (zero) through 9.

Here's a simple example. This batch file, called DIRDATE.BAT, displays a directory sorted by date and

The demo program PHONE.BAT features the 13 of DOS's 14 batch-file symbols; it lets you store and retrieve phone numbers in a database.
Line numbers are for reference only; don't type them in.

```
1 @ECHO OFF
2 ECHO.
3 IF "%1"=="" GOTO HELP
4 GOTO START
5 This batch file stores and retrieves
6 phone numbers typed at the DOS prompt.
7 :START
8 ::The following IF statements capture
9 ::information the user provides at
10 ::the command line.
11 CLS
12 IF "%1"=="/s" GOTO SORTING
13 IF "%1"=="/S" GOTO SORTING
14 IF "%1"=="/f" GOTO FINDING
15 IF "%1"=="/F" GOTO FINDING
16 IF "%1"=="/l" GOTO LISTING
17 IF "%1"=="/L" GOTO LISTING
18 IF "%1"=="/h" GOTO HELP
19 IF "%1"=="/H" GOTO HELP
20 ::
21 :ADDNUMBER
22 ::This routine adds an entry to a phone list.
23 SET FILENAME=%4
24 IF "%2"=="" GOTO HELP
25 IF "%3"=="" GOTO HELP
26 IF "%4"=="" SET FILENAME=MAINLIST.LST
27 ECHO Adding to phone list %FILENAME%:
28 ECHO.
29 ECHO %2, %1__%3
30 ECHO %2, %1__%3>>%FILENAME%
31 SET FILENAME=
32 GOTO END
33 ::
34 :SORTING
35 ::This routine sorts a phone list.
36 IF "%2"=="" GOTO SORTABORT
37 IF NOT EXIST %2 GOTO NOFILE
38 SORT <%2 >PHONE.TMP
39 COPY PHONE.TMP %2>NUL
40 DEL PHONE.TMP
41 ECHO Phone list sorted.
42 GOTO END
43 :NOFILE
44 ::This routine shows an error message
45 ::if a wrong filename is specified.
46 IF NOT EXIST %2 ECHO Invalid list name!
47 :SORTABORT
48 ::This routine displays an error message
49 ::if the user doesn't specify a phone list.
50 ECHO.
51 ECHO Proper syntax for sorting is:
52 ECHO.
53 ECHO PHONE /S [listname]
54 ECHO.
55 ECHO Your phone lists are:
56 ECHO.
57 DIR *.LST /B
58 GOTO END
59 ::
60 :FINDING
61 ::This routine searches all phone lists for
62 ::a name or number.
63 IF "%2"=="" GOTO FINDABORT
64 ECHO Finding number:
65 FOR %%A IN (*.LST) DO FIND /I "%2" %%A
66 GOTO END
67 :FINDABORT
68 ::This routine displays an error message
69 ::if the user doesn't specify a string.
70 ECHO Proper syntax for finding a number is:
71 ECHO.
72 ECHO PHONE /F [string]
73 ECHO.
74 ECHO The string can be any part of a
75 ECHO name or number. All phone files
76 ECHO are searched.
77 GOTO END
78 ::
79 :LISTING
80 ::This routine displays a phone list
81 ::specified at the command line.
82 IF "%2"=="" GOTO LISTABORT
83 IF NOT EXIST %2 GOTO NOFILE
84 TYPE %2 | MORE
85 GOTO END
86 :NOFILE
87 ::This routine displays an error message
88 ::if an improper filename is specified.
89 IF NOT EXIST %2 ECHO Invalid list name!
90 ECHO.
91 :LISTABORT
92 ::This routine displays an error message
93 ::if the user doesn't specify a phone list.
94 ECHO Proper syntax for viewing a list is:
95 ECHO.
96 ECHO PHONE /L [listname]
97 ECHO.
98 ECHO Your phone lists are:
99 ECHO.
100 DIR *.LST /B
101 GOTO END
102 ::
103 :HELP
104 ::Generic help screen
105 CLS
106 ECHO HELP SCREEN
107 ECHO.
108 ECHO To add a number:
109 ECHO PHONE [firstname] [lastname] [number]
        [listname]
110 ECHO [listname] is optional.
111 ECHO.
112 ECHO To sort the list:
113 ECHO PHONE /S [listname]
114 ECHO.
115 ECHO To find a number:
116 ECHO PHONE /F [string]
117 ECHO.
118 ECHO To list all numbers:
119 ECHO PHONE /L [listname]
120 :END
121 ECHO.                                    End
```

pauses after displaying a screen of information:

```
@ECHO OFF
DIR %1 /O:D /P
```

The value of the variable %1 depends on what you type when you execute the file DIRDATE.BAT. For instance, if you type DIRDATE *.DOC at the DOS prompt, the batch file displays all files with the extension DOC. DIRDATE is the name of the batch file; *.DOC is called a *replaceable parameter*.

When DOS reaches the second line of the batch file, it replaces the variable %1 with the parameter and runs this command:

```
DIR *.DOC /O:D /P
```

If you type DIRDATE without the parameter, DOS sees the variable %1 as empty and runs this command:

```
DIR /O:D /P
```

The number of parameters you want to make available to the user determines the number of replaceable parameters you include. Let's say you want to let the user include two parameters when executing DIR. Your batch file might look like this:

```
@ECHO OFF
DIR %1 %2 /O:D /P
```

If you then type this command:

```
PHONE *.DOC /S
```

DOS reads the second line of the batch file as this:

```
DIR *.DOC /S /O:D /P
```

and displays all DOC files in all subdirectories (/S) within or below the current directory. (Although DOS's batch language provides only ten replaceable parameters, you may overcome that limitation by using the SHIFT command, which effectively bumps each parameter down

a notch, pushing the parameter originally in %4 into %3, the parameter in %3 into %2, and so on.)

One popular trick among batch-file programmers is using variables to branch to different sections of a batch file. That is, if a variable equals one value, the batch file performs one task; if a variable equals another value, the batch file performs a different task. PHONE.BAT offers a simple example in line 3:

```
IF "%1"=="" GOTO HELP
```

I'll take a closer look at IF statements in a minute. For now, you need to understand only the statement's logic. If the variable %1 has no value, the program goes to the section labeled HELP. If the variable %1 has a value, the program executes the next line.

That is, if you type PHONE at the DOS prompt and press Enter, the batch file displays a help message and returns to DOS. But if you type PHONE followed by a parameter, %1 contains a value, and the batch file continues with line 4.

The statements starting at line 12 make further use of variables and IF statements. Each line compares the content of the variable %1, and DOS sends you to another section of the program if it finds a match.

For example, line 12 translates as follows: If you type PHONE /s, PHONE-.BAT goes to the section called SORTING and sorts your phone list. If the variable %1 contains none of the values in lines 12 through 19, PHONE.BAT executes the next line and, shortly thereafter, reaches the ADDNUMBER section, which begins in line 21.

ADDNUMBER contains several more IF tests. Lines 24 and 25 check variables %2 and %3 for values. If either is empty, the batch file jumps to the HELP routine. If both contain values, PHONE.BAT is ready to add a name and a phone number to your phone list. That is, if you type the following at the DOS prompt:

```
PHONE John Smith 555-1111
```

the batch file fills the variables %1, %2, and %3 with *John*, *Smith*, and *555-1111*, respectively.

Note that you don't have to use variables in the order in which you type the parameters at the command line. In line 29, before the ADDNUMBER section adds the number to the phone list, PHONE.BAT echoes the variables to the screen in this order:

```
Smith, John  555-1111
```

You'll notice that I'm ignoring variable %4 for now. This IF statement does something a bit different from the others, checking whether you want to add the number to the main phone list, MAINLIST.LST, or to a list you specify at the DOS prompt as a fourth parameter. I'll look at this more closely in the sections dealing with the double equal sign (==) and environment variables (%VARIABLE%).

One other note on variables: DOS reserves the variable %0 for the base of the batch file's filename. For example, if you include the command ECHO %0 in PHONE.BAT and execute the batch file by typing PHONE, the batch file displays this:

```
PHONE
```

This variable is useful in batch files that need to copy or call themselves. For example, in a batch file called MYBAT.BAT, the following line copies MYBAT.BAT to MYBAT.BAK:

```
COPY %0.BAT %0.BAK
```

Because this solution works even if you rename your batch file, it's better than the more obvious choice:

```
COPY MYBAT.BAT MYBAT.BAK
```

The Double Equal Sign (==)

The IF statement in line 3 uses a double equal sign to check the value of %1. The double equal sign and the single equal sign cause consid-

erable confusion among beginning batch-file writers. The rules are simple. Use == to compare values between two items: If ITEM1 equals (==) ITEM2, then perform an action. Use = to set the value of an environment variable: Set ITEM1 equal to the value of ITEM2.

I'll elaborate on the single equal sign in a minute. First, I must clarify a point regarding the IF statements in PHONE.BAT. You'll notice that the items on either side of the equal signs in line 3 include quotation marks; the items in lines 12 through 19 don't have to (although I've used them here to be consistent). The quotation marks in line 3

are a sneaky way to get the IF statement to check for a null value. When %1 is empty, DOS executes this statement:

```
IF ""=="" GOTO HELP
```

Because the statement is true, the batch file branches to the routine called HELP.

What would happen if you left the quotation marks off? DOS would read the statement as follows:

```
IF == GOTO HELP
```

Because this is an illegal statement, DOS would display a syntax-error message.

By the time the batch file reaches the IF statement in line 12, it has already tested %1 for a null value. Because %1 must contain something, the quotation marks are no longer needed (although, as noted above, you may include them).

Quotation marks aren't the only characters you may use in an IF statement, but they're one of the more common symbols programmers use, partly because they conform to the syntax requirements of certain higher-level programming languages. You may produce the same result with any of the following commands:

```
IF %1!==! GOTO HELP
IF '%1'=='' GOTO HELP
IF _%1==_ GOTO HELP
```

The Colon (:) and Double Colon (::)

A colon's main job is identifying a routine label. In PHONE.BAT, the first routine is START in line 7. Other routines in the batch file include ADDNUMBER (line 21), SORTING (line 34), and SORTABORT (line 47).

You go to a routine by typing a command such as the following:

```
GOTO label
```

In PHONE.BAT, line 4 jumps to the routine called START. At first glance, this section of the program seems superfluous. It's not called from anywhere else in the batch file. Why not delete lines 4 and 7 and plunge straight into the program? No reason—but it illustrates an alternative way to put comments into a batch file. Because DOS leapfrogs lines 5 and 6, they don't need REM statements. In a batch file containing numerous comments, this technique saves storage space. It also speeds execution; REM statements are commands and therefore are executed as the batch file runs, while double colons are ignored.

Contrary to the instincts of experienced DOS users, labels may contain more than eight characters. DOS looks only at the first

USING PHONE.BAT

Although I designed PHONE.BAT primarily as a teaching aid, it's not just a demonstration tool. After you finish studying how I use symbols to do what needs doing, you may put the listing to work keeping track of your phone list.

To add a phone number, use the following syntax:

```
PHONE firstname lastname number [filename]
```

The filename is optional. If you exclude it, the number is added to the main list, MAINLIST.LST. If you specify an alternate file, you must include the extension LST (for example, BUSINESS.LST or PERSONAL.LST).
Entries are stored in this order:

```
lastname firstname number
```

To find a number, use this command:

```
PHONE /F string
```

The string may be any part of a number or name, and you may use upper- or lower-case characters. PHONE.BAT searches all phone lists for the string.

To list all numbers in a file, use this command:

```
PHONE /L filename
```

The filename is required.
To sort, use this command:

```
PHONE /S filename
```

When you use /L or /S, you must include a filename.
The main article discusses symbols in the order in which they first appear in the listing. When you type in the listing, don't include the line numbers.

—E.M.

eight characters, however, when searching for a label. For example, to DOS these labels are identical:

```
FORMATTING_A
FORMATTING_B
```

Because DOS always starts at the beginning of a file when looking for a label, it will never find FORMATTING_B, even if the GOTO is closer to it than it is to FORMATTING_A.

Two colons have an entirely different function from a single colon: They replace REM. For example, a batch file might start like this:

```
@ECHO OFF
::Instructions on how to use
::the batch file
::go on these three lines.
:START
```

The main difference between using two colons and using REM is that the colons make your batch file run faster. But there's a second circumstance under which a double colon might occasionally be the better choice: when your remarks include redirection and pipe symbols. The redirection symbols are the greater-than (> and >>) and less-than (<) signs; the pipe symbol is a vertical line (|). Next time, I'll discuss all four symbols more fully.

For now, suffice it to say that programmers use these symbols to redirect data. It's a peculiarity of DOS that it recognizes redirection and pipe symbols within a REM statement. Thus, when DOS encounters the following statement:

```
REM Press <Ctrl+C> to stop the program
```

it tries to route data from a file called CTRL+C to a program called PRESS. When it doesn't find either CTRL+C or PRESS, it returns a "File not found" error message. The batch file will continue to run, but the error message can be disconcerting. To avoid the problem, use this comment line instead:

```
::Press <Ctrl+C> to stop the program
```

Or leave out the greater-than and less-than signs; most users will understand the comment if you omit them.

You now have the essentials on the "basic six." In the next issue, I'll discuss the remaining eight symbols: the equal sign, two percent signs, the double and single greater-than signs, the less-than sign, the double percent sign, the pipe symbol, and the slash. In the meantime, take a few minutes to play with the first six symbols, using them in your own batch commands or modifying lines in PHONE.BAT. ■

SYMBOL SUMMARY

Symbol	Line Number of First Reference in PHONE.BAT	Function
At sign (@)	1	Stops the remainder of the line from being displayed.
Period (.)	2	Used with ECHO, displays a blank line.
Percent sign (%)	3	Precedes a number from zero to 9; indicates a replaceable parameter, a value you set when executing a batch file.
Double equal sign (==)	3	Compares two values in an IF statement.
Colon (:)	7	Precedes a line label.
Double colon (::)	8	Precedes a remark; replaces REM.
Equal sign (=)	23	Sets the variable on the right side of a command to the value of the string or variable on the left side.
Two percent signs (%variable%)	27	Indicates an environment variable, usually set within the batch file.
Double greater-than sign (>>)	30	Adds data to a file; if the file doesn't exist, DOS creates it.
Single greater-than sign (>)	38	Reroutes data to a device or new file.
Single less-than sign (<)	38	Reroutes data from a file to a command.
Double percent sign (%%)	65	Precedes a variable in a FOR...IN...DO statement.
Pipe symbol (	)	84	Sends the output of one program as input for the next program.
Slash (/)	none	Used within a FOR...IN...DO statement, breaks strings in two.

CHOOSING A SYSTEM

How to Get What You Really Want

Whether you're considering a new, out-of-the-box system or a custom-built PC, the computer of your dreams may be more affordable than you think.

by Robert L. Hummel

I n computing, every day brings a new innovation, every innovation a new product, and every product a new standard. If you're in the market for a new PC system, the decision-making process can seem overwhelming. Important questions begging to be answered include "How do I get the most power for the least money?" and "How do I know when *good* is good enough for my needs?"

Answers to these and other significant questions begin to come into focus if you follow a basic computer buying principle: No matter whether you want to buy a ready-to-go system off the shelf or build your own new PC from the ground up, tackle your purchasing decisions one step at a time. In this article, I'll examine each major PC component and give you guidelines

Contributing Editor Robert L. Hummel is the author of PC Magazine Programmer's Technical Reference: The Processor and Coprocessor, PC Magazine Assembly Language Lab Notes, *and* Programmer's Technical Reference: Data and Fax Communications *(all from Ziff-Davis Press, 1992).*

for specifications and performance for the highest-performance and most affordable systems. (See the sidebar, "A System to Fit Your Budget," pages 42 and 43, for summaries and product recommendations. And for tips on dealing with the people behind those systems and components, see the companion article, "12 Steps to Painless Customer Support," pages 44 and 45.)

CHOOSING A SYSTEM

For a house to function as a shelter, it requires a sound foundation, good walls, and a tight roof. In the same sense, the viability of a good computer system depends on three components: CPU type, CPU speed, and system-bus structure.

Central Intelligence

The CPU, or *central processing unit*, a chip about the size of a large postage stamp, controls the operation of your computer.

The venerable 386 CPU, which opened the door for Windows' and OS/2's virtual memory management and multitasking, has been replaced by the 486 as the entry-level chip of choice and the Pentium

(a.k.a. 586) at the high end. So, you ask, with only two CPU types, how can there be such a bewildering array of computers?

The answer becomes clearer when you understand that each chip comes in a variety of *clock speeds*, which determine how fast your computer can perform multiple tasks. The 486, for example, comes in 66, 80, and 100MHz versions, while the Pentium comes in 60, 66, 75, 90, 100, and 120MHz versions.

In simplest terms, if you use a variety of business and graphics-oriented applications, the higher the clock speed, the better. By the same token, if the majority of your work is word processing, you don't need to invest in a more expensive 100MHz chip when a 60 or 66MHz would suffice.

The CPU is a key element in the *system board*, also known as the *motherboard*, which contains all the circuitry, sockets, switches, and memory needed to make your PC function. In addition to choosing the right CPU, you should also consider a system board's RAM (random-access memory) capacity, cache size, and bus structure.

Version Notation

This DOS command reference is designed for users of MS-DOS 3.3 through 6.22. To make it clear which features of a particular command you may use with your DOS version, we use the following notation system (unnotated commands work with all versions):

4 denotes a command or feature that is valid for version 4.0 and later
5 denotes a command or feature that is valid for version 5.0 and later
6 denotes a command or feature that is valid for version 6.0 and later
6.2 denotes a command or feature that is valid for version 6.2 and later

(To find out which version of DOS you are using, type VER at the DOS prompt; to reach the on-line DOS manual, typeHelp [command] from the command line or [command] /? to find the proper switch syntax..)

Conventions

Each command's key words and switches appear in uppercase, but you may use upper- or lowercase. Italics denote parameters for which you must substitute the appropriate word. When you see a command that includes the word *filename*, for instance, you must replace it with the name of the file on which you want DOS to act. You may use upper- or lowercase.

Parameters enclosed in brackets ([ ]) are optional; all others are required. An ellipsis (...) means that you may repeat the preceding parameter. When two items are separated by a vertical bar, as in +R| - R, you may use either parameter (+R or - R).

Command examples assume that the directory you use to store DOS's external files (FORMAT.COM and other files that aren't part of COMMAND.COM) is the current directory or that this directory is listed in your AUTOEXEC.BAT's PATH statement. If this is not the case, you must precede each command with the appropriate path. A path may include both a drive letter and directory name (for example, C:\DOS). Similarly, all paths acted upon by commands follow this standard format.

APPEND—Establishes a search path outside the current directory, which permits programs to locate data files.
  WARNINGS: Don't use APPEND when running Windows. Some applications will read a file from an appended directory and mistakenly write changes to the current directory. You may use /X and /E only the first time you execute APPEND. If you use /X or /E, you may not include paths on the same command line.

APPEND [*path*[;...]] [/X:ON|OFF] [/PATH:ON|OFF] [/E]

path   Specifies the directory to be appended. You may enter multiple directories, separating them with semicolons. Typing APPEND ; cancels the existing list of appended directories.
/X:ON|OFF   /X or /X:ON lets APPEND search for and run executable files, just as PATH does. /X:OFF is the default setting; it turns off APPEND's ability to run executable files. DOS 3.3 supports only /X; you may use it only the first time you execute APPEND.
/E   Creates the environment variable, APPEND, which holds a list of appended paths.
4 /PATH:ON|OFF   /PATH:ON, the default, tells DOS to search for a file even if the filename a program is searching for includes a path. /PATH:OFF disables the search path if the filename includes a path.

ATTRIB—Displays and changes a file's attributes.

ATTRIB [+R|-R] [+A|-A] [+S|-S] [+H|-H] [*path*]*filename* [/S]

+R|-R   Sets/clears the read-only attribute.
+A|-A   Sets/clears the archive attribute.
5 +S|-S   Sets/clears the system attribute.
5 +H|-H   Sets/clears the hidden attribute.
filename   Name of the file or files to be processed. May include wildcard characters.
/S   Tells DOS to act on the specified files in the current directory and all of its subdirectories.

CHDIR (CD)—Changes the current directory or tells DOS to display the name of the current directory.

CHDIR [*path*]
CD [*path*]

path   Provides the name of the directory to make current. If you don't supply a path, DOS displays the name of the current directory.

COPY—Copies one or more files from one location to another. NOTE: The switches /A and /B apply to the filename immediately preceding the switch and to any filenames that follow, until the opposite switch is encountered.

COPY [/Y|/-Y] [/A|/B] *source* [/A|/B] [+*source* [/A|/B][+...]] *destination* [/A|/B] [/V]

source   The path and filename(s) of the file(s) you want to copy. You may use wildcards to copy groups of files. Listing multiple source files separated by a plus sign (+) tells DOS to combine them into one destination file.
destination   The new location and names of the copied file(s).
/A   Denotes an ASCII text file. COPY reads disk data until it finds an end-of-file character.

/B   Denotes a binary file. COPY reads the number of bytes listed in each file's directory entry.
/V   Verifies that new files can be read.
6.2 /Y   COPY will replace existing files without prompting for conformation.
6.2 /-Y   Tells DOS to prompt you before COPY replaces an existing file.

DEL (ERASE)—Deletes (erases) one or more files.

DEL [*path*]*filename* [/P]
ERASE [*path*]*filename* [/P]

filename   The name of the file(s) to be removed. You may use wildcards and directory names to delete multiple files.
4 /P   Tells DOS to prompt you for confirmation before deleting files.

6 **DELTREE**—Deletes a directory, all of its subdirectories, and all of the files they contain.

DELTREE [/Y] *path*

path   Name of directory to delete.
/Y   Tells DOS not to issue a confirmation prompt before deleting a directory.

DIR—Displays a list of the files in a directory.

DIR [*path*][*filename*] [/P] [/W] [/A[[:]*attributes*]] [/O[[:]*sortorder*]] [/S] [/B] [/L] [/C]

filename   Names the file(s) to list. Wildcards and directory names are permitted.
/P   Pauses after the screen is full. You must press a key to continue.
/W   Displays files in a wide format.
5 /A[[:]*attributes*]   Displays only files with the specified attributes. Using /A without attributes displays all files, including hidden and system files. Preceding an attribute with a minus sign (-) means *not*.

  H|-H   Displays hidden files or files that are/aren't hidden.
  S|-S   Displays system/nonsystem files.
  D|-D   Displays only directories/files.
  A|-A   Displays files ready for backup/files that haven't changed since the last backup.
  R|-R   Displays read-only files/files that aren't read-only.
5 /O[[:]*sortorder*]   Specifies the order in which you want files and directories displayed. Using /O without providing a sorting order tells DOS to display directories first and then files, listing them alphabetically. Sorting order is ascending (*A* precedes *Z*), unless a sorting-order flag is preceded with a minus sign (-), which means descending order.
  N|-N   Lists in alphabetic/reverse alphabetic order.
  E|-E   Lists in alphabetic/reverse alphabetic order by file extension.
  D|-D   Lists in ascending/descending order by date and time.
  S|-S   Lists in ascending/descending order by file size.
  G|-G   Lists directories/files first.
  6 C|-C   Lists in ascending/descending order by compression ratio.
5 /S   Provides a directory listing for the current directory and all subdirectories below it.
5 /B   Omits heading and summary information, displaying only file or directory names and extensions.

5 /L   Displays in lowercase letters an unsorted list of filenames and directories.
6 /C   Includes compression ratio in the directory listing.

4 **DOSSHELL**—Starts DOS's menuing system (shell).

DOSSHELL [/T[:res[n]]] [/G[:res[n]]] [/B]

5 /T   Starts the DOS shell in text mode. Can't be used with the /G switch.
5 /G   Starts the DOS shell in graphics mode. Can't be used with the /T switch.
5 res Sets screen resolution. Substitute l (low), m (medium), or h (high) for res.
5 n   The subcategory number of the screen resolution mode. The valid range for n varies with your computer's hardware.
/B   Starts the DOS shell in black-and-white mode.

FASTOPEN—Provides a cache for storing a disk's directory information. Using this command may decrease the time DOS spends finding frequently used files.

FASTOPEN drive:[=n] [...] [/X]

drive: The logical drive letter of the drive you want to cache. You may list more than one drive.
[=n]   The maximum number of files cached for the specified drive. The default is 34 (48 in DOS 5 and later); the range is 10–999.
4 /X   Stores the directory cache in expanded memory.

FC—A file-comparison utility that tells you how individual files or file sets differ. Files are assumed to be in ASCII unless they have EXE, COM, SYS, OBJ, LIB, or BIN extensions.

For ASCII Files:

FC [/A] [/C] [/L] [/LBn] [/N] [/T] [/W] [/lines] [path1]filename1 [path2]filename2

/A   Provides an abbreviated list of differences.
/C   Ignores the case of letters during the comparison.
/L   Compares files line by line.
/LBn Sets the number of lines for the internal buffer. If more than that number of lines differ, the comparison ends. The default value is 100.
/N   Displays line numbers during the comparison.
/T   Doesn't expand tab characters to spaces before making the comparison. By default, tabs are expanded to spaces, with tab stops every eight characters.
/W   Compresses white space (tabs and spaces), treating tabs and consecutive spaces as a single space. With this switch in effect, FC ignores white space at the beginning and end of lines.
/lines Specifies how many lines must match before FC considers two files resynchronized.
filename1 The name of the first file to compare.
filename2 The name of the second file to compare.

For Binary Files:

FC /B [path1]filename1 [path2]filename2

/B   Indicates that files being compared are binary. Files are compared byte by byte. This is the default setting for files with EXE, COM, SYS, OBJ, LIB, or BIN extensions.

MKDIR (MD)—Creates a directory.

MKDIR path
MD path

path   The location and name of directory to create.

6 **MOVE**—Moves one or more files from one location to another, or permits you to rename a directory.

MOVE [/Y|/-Y] [path]filename [,...] destination

filename   The name of the file or files you want to move, or the name of the directory you want to rename.
destination The new location or new directory name. If you are moving only one file, the destination may be a filename.
6.2 /Y   MOVE will replace existing files without prompting for conformation.
6.2 /-Y Tells DOS to prompt you before MOVE replaces an existing file.

RMDIR (RD)—Removes an empty directory.

RMDIR path
RD path

path   The location and name of the directory to be removed. The specified directory must be empty.

RENAME (REN)—Renames a file or group of files.

RENAME [path]filename1 filename2
REN [path]filename1 filename2

filename1   The original filename(s).
filename2   The new filename(s).

REPLACE—Updates or replaces existing files with more current versions of the files.

REPLACE [path1]filename1 [path2]filename2 [/A] [/P] [/R] [/S] [/W] [/U]

filename1   The source filename.
filename2   The destination (target) filename.
/A   Copies to the target all source files that don't already exist in the target. Can't be used with /S or /U.
/P   Prompts you for confirmation before overwriting files in the target.
/R   Replaces read-only files. If you omit this switch, DOS doesn't replace read-only files.
/S   Searches all directories of the target for files matching the source filename.
/W   Pauses, waiting for you to insert the source disk.
4 /U   Updates only the files in path2 that are older than those in path1.

TREE—Displays a tree-like view of a disk's directory structure.

TREE [path] [/F] [/A]

path   The drive containing the disk whose directory you want to display. In DOS 4 and later, you also may specify the name of the directory.
/F   Includes filenames in the directory display. Without /F, only directory names are listed.
4 /A   Tells DOS to use text characters (rather than graphics characters) to create the lines linking the branches of a directory tree.

5 **UNDELETE**—Recovers files deleted with the DEL command.

UNDELETE [path][filename] [/DT|/DS|/DOS]

[path][filename] The location and name of the file(s) to recover. By default, the command recovers all files in the current directory.
/DT   Recovers only files stored in the deletion-tracking file.
6 /DS   Recovers only files stored in UNDELETE's \SENTRY directory.
/DOS Recovers only files that DOS lists as deleted.

UNDELETE [/LIST|/ALL|/PURGE[drive]|/STATUS|/LOAD|/UNLOAD|/S[drive]| /Tdrive[-entries]]

/LIST   Lists all deleted files that can be recovered, but doesn't take any action on those files.
/ALL   Recovers deleted files without asking for confirmation for each file.
6 /PURGE[drive] Deletes the contents of \SENTRY.
6 /STATUS Displays the type of deletion protection that is in effect on each drive.
6 /LOAD   Loads UNDELETE's memory-resident program into memory.
6 /UNLOAD Unloads UNDELETE's memory-resident program.
6 /S[drive] Activates UNDELETE's deletion sentry.
6 /Tdrive[-entries] Activates UNDELETE's deletion tracker. The entries option indicates the maximum number of entries that the deletion-tracking file may contain.

XCOPY—An expanded version of the COPY command that lets you duplicate files and directories.

XCOPY [path1]filename1 [path2][filename2] [/Y|/-Y] [/A|/M] [/D:date] [/P] [/S[/E]] [/V] [/W]

filename1   The path and filename(s) from which to copy.
filename2   The path and filename(s) to which to copy. If you omit a destination, DOS works on the current directory.
/A   Copies only files whose archive bit is set. Doesn't change the archive attribute.
/M   Copies only files whose archive bit is set, turning off the archive bit for each file as it is copied.
/D:date   Copies only files saved on or after the specified date.
/P   Prompts you for confirmation before copying each file.
/S   Copies the current directory and the subdirectories below that directory, provided those subdirectories contain files.
/E   Copies any subdirectories, even if they are empty. Valid only if you use the /S switch.
/V   Verifies that the destination is readable.
/W   Waits for you to respond to a prompt before beginning to copy files.
6.2 /Y   XCOPY will replace existing files without prompting for conformation.
6.2 /-Y Tells DOS to prompt you before XCOPY replaces an existing file.

The original IBM PC system board could hold only 64K of RAM, configured as four banks of nine individual DIP (dual in-line package) chips. Today's systems rely on SIMMs (single in-line memory modules)—small circuit cards that plug into slots on the board. SIMMs come in 30-pin and 72-pin versions.

Although neither version is better, it's important that you match the type and speed of your SIMMs with the PC's system requirements if you're buying a new PC part by part.

Any new system should have a minimum capacity of 32MB of RAM. For a system that uses 30-pin SIMMs, for example, eight slots are arranged as two banks of four. All chips in the same bank must be of the same type, which means that if you're upgrading to a PC with 8MB of RAM, you'll get eight 1MB SIMMs.

If your budget allows, look for 16MB (arranged as four 4MB chips); it will give you improved performance and at the same time leave room for an upgrade later.

To avoid slowing down the processor and to improve overall performance, a small amount of very fast memory, called a *cache*, is added to the system board. Some system-board manufacturers try to save you money by including a cache of only 64K or 128K. But that's simply too small. Insist on a cache size of at least 256K.

By the way, don't confuse the system-board cache with a cache that's built into the CPU itself. Both the 486 and Pentium contain an internal cache to speed up operations inside the chip only. An internal cache is *not* a substitute for a system board's external cache.

Hop on the Bus, Gus

Another key feature to consider when buying any new PC is its *system bus*. In essence, a system bus lets you add sound, fax, or a scanner simply by opening the computer and popping in a new expansion board. Several system-bus types have become popular, the three most significant being VL (VESA local), PCI, and ISA.

Of the three, PCI should be your bus of choice for the present and future. Developed for the Pentium chip, PCI has also been quickly adopted for 486, PowerPC, and other systems. It provides essentially the same performance as the VL bus, which was developed for 486 machines, but will eventually dominate the market, replacing both VL and ISA-based systems.

Until that happens, however, rest assured that all VL and PCI systems now on the market also support ISA: a good thing, because nearly every expansion board is still sold in an ISA-compatible version.

> ### Buyer's Guide Summary: System Features
> - 486DX2-66 minimum
> - 256K or more cache memory
> - 4MB RAM minimum; 8MB or more recommended
> - PCI bus preferred; VL bus acceptable

CHOOSING A MONITOR

One of the most frustrating and difficult challenges you'll face when buying a new PC is choosing a monitor. In addition to mastering a bewildering array of specifications, buzzwords, and higher math, you have to be aware that although some brand-name machines are available with monitors, many IBM, Compaq, and NEC models, for example, require you to choose a monitor separately. Fortunately, you need to know only four key specifications—*dot pitch*, *resolution*, *refresh rate*, and *size*—to determine how well any monitor will perform.

Keep It in Focus

A monitor's *dot pitch* is the minimum distance between two distinct dots of light produced on screen and is measured in fractions of a millimeter. The smaller the dot pitch, the sharper an image will appear—ideal if you're displaying complex graphics such as CAD (computer-assisted design) drawings.

For most applications, look for a minimum dot pitch of .28mm. You may find a bargain monitor with a dot pitch of, say, .39mm. But don't be fooled by the price—though just .11mm bigger, even large text will look blurred, as though it's behind a piece of waxed paper.

Most monitor manufacturers express *resolution* in a shorthand notation of dots by lines (the number of dots that can fit on a single line and the number of rows that will fit on the entire screen).

Standard VGA graphics resolution, for example, is 640 by 480. Super VGA resolutions can include 800 by 600, 1024 by 768, 1280 by 1024, and 1600 by 1280.

COLOR DEPTH		
RESOLUTION	1MB VIDEO RAM	2MB VIDEO RAM
640x480	16.7 million	16.7 million
800x600	65,000	16.7 million
1027x768	256	65,000
1280x1024	0	256

The number of colors (depth) a video card can display depends directly on the amount of video RAM the card contains.

A SYSTEM TO FIT YOUR BUDGET

No matter how much or how little you have to spend, with careful shopping and exhaustive mail-order comparisons, you can come close to matching the following basic packages (except where noted, peripherals not included) within their estimated total prices. In each example listed in the left-hand column, systems reflect best performance for the money and support DOS, Windows, and multimedia applications. For perspective, and because of its renowned customer service, we've included in the right-hand column a representative tabulation from the April/May 1995 catalogue of PC Connection, based in Marlow, New Hampshire (800-800-1111, 24 hours a day, seven days a week). Note that MDS is the company's own Marlow Data Systems, and that an asterisk indicates that the particular feature is included with the main system or multimedia kit. Also, the Cornell system, as noted, isn't available from PC Connection.

THE $1000 TO $1699 PC

486DX2-66 CPU	MDS DX2/66 Desktop ($1099)
System board with VESA (VL) local bus	*
4MB RAM system memory, 256K cache memory	*
400MB IDE hard drive	425MB
VESA (VL) local-bus IDE controller	*
VESA (VL) local-bus Super VGA card with 1MB RAM	*
14-inch SVGA 1024x768 noninterlaced .28mm dot-pitch monitor	Magnavox CM2099 14-inch ($269.95)
Double-speed CD-ROM drive	Panasonic Multimedia Bundle ($249.95)
16-bit sound card and speakers	* headphones only
	Sony SRSPC30 speakers ($64.95)
1.44MB floppy drive	*
2 serial, 1 parallel, 1 game port	*
101-key keyboard, mouse	*
DOS/Windows or OS/2 Warp	DOS 6.2/Windows for Workgroups 3.1
	TOTAL: $1683.85

THE $1700 TO $2499 PC

Pentium-60 or 486DX4-100 CPU	NEC 5010 Multimedia MiniTower ($1699)
System board with PCI local bus	*
8MB RAM system memory, 256K cache memory	*
700MB IDE hard disk	540MB (730MB, $149)
PCI local-bus IDE controller	*
PCI local-bus Super VGA card, with 1MB RAM	*
15-inch SVGA 1280x1024 noninterlaced 28mm dot-pitch monitor	MAG DX15 ($379)
Quad-speed IDE CD-ROM drive	*
16-bit sound card and speakers	* includes 16 software titles, fax/modem
1.44MB floppy drive	*
2 serial, 1 parallel, 1 game port	*
101-key keyboard, mouse	*
DOS/Windows or OS/2 Warp	DOS 6.2, Windows 3.1
	TOTAL: $2277

What's important to keep in mind is that finer (as in 1600 by 1280) isn't necessarily better. For example, most new monitors support 1280-by-1024 resolution for today's multimedia and graphics-oriented applications. That sounds great, until you try to type something in Windows and find yourself squinting at very tiny words on a very small screen.

Not surprisingly, then, the size of your monitor (and your wallet) will determine how much resolution you can use productively. For example, if you're considering a monitor with a resolution of 1024 by 768, it would be uncomfortably small on an economical (around $250) 14-inch screen. The same resolution, however, would be a great fit for a much more expensive (around $650) 17-inch monitor. If you can't spend that much, the smallest practical screen for 1024-by-768 resolution is a 15-inch monitor (around $325).

But no matter what resolution you desire, the rule of thumb is this: Don't pay extra for resolution you'll never need.

Get the Whole Picture

Once you know a monitor's dot pitch and resolution numbers, the next question is "How fast is the refresh rate?"

A monitor's *refresh rate* determines how quickly it restores the intensity of dots as they appear and then fade on your screen. Phosphor dots on the screen glow as they're struck by the electron beam inside the monitor. As soon as the beam moves away, though, the dots begin to fade. If the monitor's refresh rate is fast enough, the gradual fading of individual dots is inconspicuous. But if it isn't fast, by the time the last line of the image is on the screen, the top line will have faded noticeably.

To combat this problem, some monitor manufacturers employ a technique known as *interlacing*. Instead of drawing every line of the image during each vertical scan

(noninterlaced mode), an interlaced monitor draws only every other line, taking two passes to draw a single image. The two sets of lines are overlaid to create a single image every two passes.

Interlaced monitors are usually slightly cheaper than noninterlaced, but even so, the shifting brightness of alternating lines is known to cause an annoying flicker. Not everyone is sensitive to this flicker, nor is it always perceptible, but you'll be better off choosing a monitor that provides the resolution you want at a noninterlaced rate of 60Hz or greater.

Now all you have to do is pick a *video card*, which controls your monitor's *color depth* (the number of colors you can display at one time). The accompanying table (page 41) shows the color depth that can be supported by 1MB and 2MB of video RAM at four popular resolutions.

The most important consideration when selecting a video card is to make sure that it matches your monitor's specifications. For example, if your monitor supports 1024 by 768, you'll need to make sure your video card does, too. Avoid the temptation to overbuy: Paying a premium for a high-performance card is a waste if your monitor can't support high-resolution modes.

Buyer's Guide Summary: Monitors

- .28mm or smaller dot pitch
- 1024 by 768 or better noninterlaced resolution
- 15-inch flat or full-view screen, minimum
- 1MB video card, minimum; 2MB for higher resolution

CHOOSING A HARD DISK

Hard drives have come a long way since I plunked down a month's salary for my first 20MB of magnetic real estate. Although some things haven't changed through the years—bigger is still better— today you'll be pleasantly surprised to know that bigger is also less

A SYSTEM TO FIT YOUR BUDGET CONTINUED

THE $2500 TO $3499 PC

Pentium-100	MDS 18745 Mid-Tower ($2099)
System board with PCI local bus	*
16MB RAM system memory, 256K cache memory	*
1GB IDE hard disk	1.2GB ($269)
PCI local-bus IDE controller	*
PCI local-bus Super VGA card with 2MB RAM	*
15-inch SVGA 1280x1024 noninterlaced 28mm dot-pitch monitor	MAG DX15 ($379)
Quad-speed IDE CD-ROM drive	Toshiba 4X ($349.95)
16-bit sound card and speakers	Sound Blaster 16 Value Edition ($99.95) Sony SFS-D2KPC speakers w/subwoofer ($149.95)
1.44MB floppy drive	*
2 serial, 1 parallel, 1 game port	*
101-key keyboard, mouse	*
DOS/Windows or OS/2 Warp	DOS 6.2, Windows for Workgroups 3.1
	TOTAL: $3346.85

THE $7000 TO $8999 "DREAM MACHINE"

Pentium-100	** Cornell Dual PCI/EISA Ultima Workstation ($4595)
System board with PCI local bus	*
32MB RAM system memory, 256K cache memory	*
2GB Wide SCSI-3 hard drive	2.1GB ($1099)
PCI local-bus Wide SCSI-3 controller	*
PCI local-bus Super VGA card with 4MB RAM	*
21-inch SVGA 1600x1280 noninterlaced .25mm dot-pitch monitor	NEC XE21 ($2079)
Quad-speed SCSI-2 CD-ROM drive	Toshiba 4X ($349.95)
16-bit wave-table sound card, speakers	Sound Blaster 16S ($179.95) Altec Lansing speakers w/subwoofer ($219.95)
1.44MB floppy drive	*
2 high-speed serial, 1 parallel, 1 game port	*
101-key keyboard	Ergo Enhanced ($69.95)
Mouse	Logitech Expert 4.0 ($114.95)
OS/2 or Windows NT	Windows NT Workstation 3.5 on CD-ROM ($295.95)
	TOTAL: $8603.70

** not available from PC Connection; call Cornell Computer Systems (800-989-3475)

PROCESSOR NOMENCLATURE

Processors are described by their type and speed in a sort of shorthand notation. The first part of the notation describes the generic processor family, such as 486 or Pentium. The *DX* is the dominant variant of the 486, and simply means that the chip runs twice (DX2) or four times (DX4, or 4DX) as fast internally as externally. The final part—say, the *66* in a 486 4DX-66 notation—is the chip's internal clock speed in megahertz (MHz).

—R.H.

12 STEPS TO PAINLESS CUSTOMER SUPPORT

Let's face it: Too many of us approach calling a computer customer-support number with the same enthusiasm we reserve for tax audits and dental surgery. At least with the latter, you might be offered laughing gas.

Too often, a call to customer service or technical support can leave you frustrated. Novices can get lost easily, while seasoned veterans wanting a quick solution find themselves listening to an involved explanation of the functions of AUTOEXEC, all of which may follow a long course through busy signals and hold patterns.

Is it actually possible to maximize your success rate when you need support, while you minimize time and money spent trying to get that help? After talking with several industry experts, I'm happy to say that the answer is yes.

Make your first call the only call. Ideally, the best way to determine any company's customer support is to check it out before you purchase its product. Get the manufacturer's support number, try calling it at a time of day when you're apt to make your support calls, and note how long it takes to get through, how friendly and knowledgeable the response is when you get it, and how direct are answers to questions about the company's support and warranties. Determine whether the company offers on-site support, what its return policy is, and whether the computer or other product you're buying comes with industry-standard components or proprietary ones that may be harder to replace.

It's also wise to ask how long the company can guarantee the availability of replacement parts. If it can't guarantee availability and the unit has a lot of proprietary components, you could be looking at a future of many long, unsatisfying calls in search of support.

Although checking out customer support before you buy is the best consumer safeguard, there are several tips for succeeding even if you've already bought new hardware or software:

1. Before you call, become familiar with the product's manuals and any on-line help if available. Many manuals include a list of troubleshooting techniques, and it costs less to read them yourself than to have a support technician read them to you.

2. Know the brand, model name, CMOS settings and BIOS date of your computer, and with software or hardware, know the basics of the computer on which you've experienced the problem. You should also write down what version of DOS and/or Windows you're using, and complete information on any disk-compression utility. Many support lines also require a product's serial or registration number before you can actually speak to a real person. So get a small notebook and record all vital system information. If you're unsure about your system's technical make-up, use any system-information reporting program (MSD, Check-It, or Norton's SysInfo, for example) to check your setup, then jot down the relevant information.

3. If your problem seems to crop up "out of the blue," be sure to jot down all information about what you were doing immediately before trouble developed. Then try to backtrack to see whether there was anything you may have changed in your system in the past few weeks. For example, you might not think that the new font manager you installed last month would suddenly cause Windows to hiccup three weeks later, but anything is possible.

4. When it's time to make the call, take charge of the conversation; be assertive yet calm, and fully prepared to ask questions. Your attitude when you call could make or break your success with customer support, so if you're feeling frustrated and angry from hours of troubleshooting, call later.

5. Make certain you and the support operator are talking about the same thing. Don't pretend to understand something you don't simply to expedite the phone call; after all, no one knows everything.

6. Take notes, and, as soon as instructions or explanations stop making sense, tell the support person to explain.

7. Ask whether you can try possible fixes while you're on the line; if that's not possible, ask for an incident number or some way of tracking the ground you've already covered. Many companies record calls in a database so that they can bring up your system information quickly on future support calls. If that's not available, ask whether there's a way to call back directly,

---

expensive. As I write this, a price of 34 cents per megabyte is common, and dropping daily.

One thing that has changed over the years, however, is hard-drive standards. While several have come and gone, today you need to consider only two families—IDE and SCSI—when shopping for a new computer.

Dumb No Longer

The earliest hard-disk drives were "dumb." All the intelligence needed to operate them was located on the drive's controller expansion card. That type of design made hard disks cheaper, certainly, but it also created a serious problem: If your controller card didn't support your drive, you couldn't use it.

Because drive manufacturers quickly realized that drives that can't be used don't sell, they developed what's known as the *IDE* (Integrated Drive Electronics) system, in which most of the controlling electronics are built into the drive itself. For quite some time, IDE has been popular, practical, and pervasive. But because it's

or whether there's another knowledgeable person for whom you can ask by name.

8. Sometimes, no matter how prepared you are, you'll hit the wall just trying to get through to someone—anyone. Say you're trying to reach tech support but fail repeatedly. Try calling the customer-support number, explain the difficulties you're experiencing trying to get through, and ask for assistance. Sometimes they can patch you directly through to a technician.

9. If you get through but feel unsatisfied with the answer, ask for a senior technician. "Avoid asking for a supervisor just because you're angry," advises Robin Bush, a Richmond, California-based PC consultant and on-line technical manager. "Use it for those situations when you feel the support rep knows less about your problem or computers than you do."

10. Because many companies hire inexperienced outside personnel to answer questions from a prepared handout, you may end up going nowhere with your problem. Put aside your frustration (the lawsuit will have to wait until you've gotten your computer to boot again) and investigate alternative avenues for help. One burgeoning resource is manufacturer-support forums on major on-line services such as CompuServe and America Online, as well as the Internet, where you can locate bug fixes, technical notes, FAQs (frequently asked questions), drivers, and updates, and can often get a response from a technician within 24 hours. "On line is the ideal way for me to get support," says John Lehr, a New Jersey network supervisor. "Calling an 800 support number is like Russian roulette: You get whoever picks up the phone first. On line I can tap the minds of dozens if not hundreds of other users and support personnel pretty quickly. E-mail is better, too, for communicating technical problems and solutions."

11. If all else fails, try faxing—as long as you're not in a hurry for an answer. Since almost everyone can now afford fax technology, the number of fax inquiries to tech/customer-support lines has grown sharply. You may have to wait as long or longer for a fax-back response as you would trying to call by phone.

12. Whether your experience is favorable or unfavorable, get the product-support director's name so that you can drop him or her a note. Include what you found useful or unsatisfactory about the level of help you got; this type of follow-up helps support departments determine what they'll need to do to provide quality customer assistance in the future.

There's hope on the horizon. The last few years haven't always been kind to either the computer industry or the users trying to get support for their products, but that could all be changing for the better.

As companies needed to grow leaner and meaner, 800 numbers and free support options began to disappear, while 900 numbers and "I'll need a major credit card first" options sprang up. The tide may have turned, however, with the swelling number of frustrated computer owners and users who complained loudly, certain the industry had developed a jaded attitude toward their problems.

In late 1994, for example, when the Pentium-chip flak drew so much ire from the computing public, Intel responded by making exchanges easier to obtain. Some industry analysts believe that Intel's repentance over the Pentium issue may have pushed Microsoft into announcing the delay of Windows 95 to ensure greater stability before its release (a point, by the way, that Microsoft has steadfastly denied).

Many leading computer manufacturers—including Hewlett-Packard, Compaq, Dell, Gateway, and Zenith—also chose 1994 to introduce three-year warranties on their desktop PCs, giving consumers longer coverage time to get assistance.

And 1994 was also the year that on-line technical support began to flourish, with help just a modem dial away.

But even a modem and extended warranties won't cure all your customer-support headaches. If frustration from trying to run a new system or software gets you down, don't compound your problems by ignoring customer-support safeguards. Follow the Scouts' advice: "Be prepared" in every way before you call.

—**Kate Chase**

---

limited to supporting drives with a maximum of 504MB, a new standard—*SCSI* (Small Computer Systems Interface)—has come into play for bigger drives.

The SCSI standard differs considerably from IDE or serial-port interfaces, in that it defines a system in which intelligent peripherals (such as a CD-ROM drive) exchange commands and data over a private communications channel known as a *SCSI bus*.

Because the SCSI bus resembles a small network, adding such intelligence to a peripheral makes it costlier to manufacture. Accordingly, then, SCSI devices typically cost more than their "dumber" IDE counterparts.

With ROM-based extensions available on SCSI adapter cards, however, a DOS system can break through IDE's 504MB limit and access hard disks as large as a whopping 8GB. In addition, a SCSI bus boasts a data-transfer rate of 5MB to 40MB per second, while the upper limit for IDE data transfer is 4MB per second.

But Wait, There's More

As demand for disk space has grown, SCSI has become the interface of choice. Recently, however, two independent (and practically identical) extensions to the IDE standard have been announced. One is called *AT Attachment 2* (ATA-2), and the other *Enhanced IDE* (EIDE). Both support IDE drives up to 8GB, and transfer rates as high as 16MB per second.

IDE power users wanting to take advantage of EIDE will have to purchase an EIDE-compatible controller (for an additional $30 to $50). You'll gain improved data-transfer speed, the ability to access EIDE drives over 504MB as a single volume, and 32-bit disk and file access under Windows.

The real question then becomes: How much hard disk do you really need? For perspective, consider the following: If you want to install the Microsoft Office suite, plan on chewing up 110MB. If you add Windows, plan on a 30MB to 50MB chunk for its permanent swap file. And if Windows 95 or OS/2 is in your future, count on another 50MB. Bottom line: No new system should come with a hard disk smaller than 400MB.

Buyer's Guide Summary: Hard Drives

- Get true drive size for comparisons.
- Drives from 500MB to 800MB offer the best value.
- Drives larger than 504MB require a special controller.

CHOOSING PERIPHERALS

After you've chosen the processor type, hard drive, and video system for your new PC, you'd like to think that the hard work is behind you. Raw power is nice, but it's the accessories and peripherals that turn your computer into the versatile workhorse you expect today's PC to be. Among the most popular and beneficial add-ons are modems, CD-ROM drives, sound cards, and tape backups. Here are some concise guidelines to consider before you plunk down your money.

Watch Your Speed

Today's top-of-the-line modems conform to the V.34 specification, which means they're speedier than ever—boasting a raw transfer rate of up to 28.8 kilobits per second (kbps), and a compressed transfer rate of up to 115kbps. What's even more remarkable is that you can get a modem with fax capability for about $200.

You can find cheaper modems, but they often skimp on the extra circuitry needed to ensure reliable communications. For example, a bargain-basement 28.8kbps modem that can connect dependably only at a slower speed is money down the drain. Regardless of your budget, under no circumstances should you settle for a modem that's slower than 14.4kbps.

Double or Quad CD-ROM?

The past few years have seen a slow but steady decrease in the price of CD-ROM drives, while performance has increased with double, triple, and now quad-speed models. Better performance has also required a separate controller card, resulting in difficult installation.

Recently, however, Mitsumi Electronics Corporation (6210 North Beltline Road, Suite 170, Irving, TX 75063; 214-550-7300) introduced its FX-400 quad-speed CD-ROM drive with IDE interface for less than $200. With little effort you can install it in your system, connect it to the same cable that attaches to your IDE hard disk, load the drivers, and go.

DOS TIP: DISABLING LETHAL COMMANDS

Reducing your chances of accidentally losing data can be as easy as deleting DOS programs such as FORMAT.COM and DELTREE.EXE from your hard disk. The trouble is, of course, that many dangerous commands are useful when handled correctly. You can rename them, but that may increase the chance that you'll run them by accident. A better choice is using one of DOS's quirks to disable those commands while retaining access to the programs.

When you type a command at the DOS prompt, DOS always checks its table of internal commands (commands such as DIR, which are built into the DOS kernel) before looking for external files (the programs and utilities DOS stores in your C:\DOS directory). If DOS finds a match, it executes the internal command. Thus, if you name a program DIR.COM, you can't execute it by typing DIR at the command prompt, because DOS intercepts the request and displays a directory listing.

You may use this property to "hide" a command and yet leave it in plain sight. For example, I've renamed DELTREE.EXE as CHDIR.EXE. When my kids type DELTREE nothing catastrophic happens; instead DOS displays a "Bad command or file name" message, because DELTREE.EXE doesn't exist. If they type CHDIR, DOS simply displays the current directory, because CHDIR is an internal command.

When I want to access the DELTREE command, I use another trick. I preface the command name with a directory reference such as this one:

```
C:CHDIR
```

That tells DOS to skip over its internal command list. Note that the file CHDIR.EXE doesn't have to be on drive C. As long as it's mentioned in your PATH statement, DOS will find it.

—Robert L. Hummel

But before you rush out to buy this revolutionary drive, you'll need a dual connector cable and a dual-channel IDE board (about $30). Without the right cable and board, you won't be able to use 32-bit disk access under Windows and make connections with the CD-ROM drive in the same system.

Despite its foibles, an IDE CD-ROM drive gives you state-of-the-art performance for mere pennies on the dollar. But if your budget is especially tight, you can still take part in the multimedia revolution by purchasing a double-speed drive for about $100.

Sound Investments

Whether you're interested in using speech or importing music, adding sound to your PC can be a great investment—as long as you know what type of sound card is best for each application.

If you intend to work with speech only and are willing to accept marginal quality, a relatively inexpensive 8-bit card with 11kHz sampling rate will be sufficient. But if you want to enjoy multimedia software or perhaps reproduce CD-quality audio from WAV and MIDI files, your card must be 16-bit with a 44kHz sampling rate.

An extra benefit is that a 16-bit card can also digitize and reproduce 8-bit sound, so it's perfect for either speech or music.

The current crop of inexpensive ($100 or less, including speakers) 16-bit sound cards can reproduce the sound of different instruments via a technology called *FM synthesis*. But the hottest trend in PC music is the 32-bit *wave-table* sound card, which contains digital samplings of actual instruments stored in ROM. The increase in quality is clear, but only you can decide whether it's worth the extra $200 to $250.

Just in Case . . .

Whether you buy piecemeal or out of the box, your new PC will likely come with DOS 6.0 or 6.2, Windows 3.1 or Windows for Workgroups, and a spate of utilities loaded on the hard disk. To save money, many dealers no longer include disk copies of each program, so the first job you'll have is to back up your system.

As hard disks and applications have grown, however, backing up with floppies is no longer practical. For a cost-effective alternative, consider a tape drive (about $150) that uses miniature cartridges and holds about 120MB of files. With compression, a tape drive can often squeeze as much as 250MB of files onto a single tape. Installation is a snap, because the drive attaches to your existing floppy controller. (For advice on developing a strategy for safeguarding your programs and data on tape, see "A Backup Blueprint" in this issue, page 23.)

Pointing and Pecking

Because you're either destined to spend a lot of time pecking away at your keyboard, or mousing around if you run Windows programs, it's important not to underestimate the value of a quality keyboard and mouse.

Unfortunately, most desktop systems come with cheap keyboards, whose mushy keys can drive a touch typist mad within a few hours. IBM has long had a reputation for the best keyboards in the business, and it's well-deserved. In fact, I picked up several genuine IBM keyboards at a local computer flea market to sock away for the future. Focus and Northgate also make quality keyboards in several varieties.

The terrors of a substandard mouse are even more insidious. I have one mouse, included with an otherwise quality system, that ranks as the worst I've ever used. The top is arched, but not like any human hand I ever saw. The edges are sharp, and dig into your palm after a few minutes of use. Clicking the buttons takes more effort than popping the top on a soda can. I can barely operate it; my kids won't even try.

So comfort is essential when choosing a mouse. Microsoft's mice (around $60) boast quality, and lesser-known but acceptable varieties are available from other manufacturers for $20 or under. It's wise to stick with a Microsoft-compatible device for easiest installation and compatibility with most applications.

MINIMIZING PROGRAMS WITH MINIMAL EFFORT

Window TIP

Are you wasting time and mouse clicks? You are if you start minimized versions of your Windows programs by double-clicking on the appropriate program icon, waiting for the program to load, and clicking on the Minimize button. To handle this operation in one step, hold down the Shift key when you double-click on an icon. In a few seconds, Windows will deposit the program's minimized icon along the lower edge of your desktop.

Note that this technique may not work with all DOS programs, because not all DOS programs work the same way. For example, different programs ask for expanded or extended memory in various ways, and your system also loads and shares memory-resident programs differently.

—Lenny Bailes

Buyer's Guide Summary: Peripherals

- modem: V.34 28.8kbps with fax
- CD-ROM drive: quad-speed IDE
- sound card: FM synthesis, 16-bit
- tape backup: 250MB or better
- keyboard: 101-key, IBM
- mouse: Microsoft-compatible, serial ■

INTRODUCING OS/2 3.0

Warp Speed for DOS

Tired of waiting for DOS 7.0 or Windows 95? IBM's OS/2 Warp is a powerful alternative, offering true multitasking, built-in memory management, and nearly infinite customization.

by Stan Miastkowski

I f you're the type of person who likes to push DOS to its limits, IBM's OS/2 Warp can open up new horizons in your computing experience. Warp—IBM's prerelease code name for its OS/2 version 3.0—is a powerful 32-bit operating system (you'll need a 386, 486, or Pentium) that can run multiple OS/2, DOS, and Windows applications at the same time.

The Basics

Think of Warp as the ultimate DOS utility set: You'll be able to eliminate a shelf full of add-in software you'd usually need to maximize DOS, such as memory managers (MemMaker, QEMM, NetRoom, or 386MAX), caching programs (SmartDrive), and multitaskers (DESQView or Windows). With a suggested price of $80, Warp has them all and more—including Internet access—built in.

*Stan Miastkowski has been translating computer technology into English for the past 17 years, and has been an OS/2 user since version 1.0. Contact him on the Internet at **stanm@bix.com**.*

With an all-in-one system like this, however, comes a more time-consuming installation. For example, instead of the three disks you'd use to install MS-DOS 6.22, the OS/2 Warp box contains 25 disks. (If you're fortunate enough to have a multimedia PC, a less cumbersome CD-ROM installation is available.) Although you won't need them all for a typical installation, plan on a couple of hours of disk swapping—and by all means, make a backup before you start.

A typical Warp installation will take up to 30MB of hard-disk space, compared with about 5MB for DOS. To play it safe, your hard disk should have a total capacity of no less than 120MB to 200MB. (See the sidebar "Stacker Squeezes Surplus Space," page 52, for information on using hard-disk compression with Warp.)

Additionally, you'll need at least 8MB of RAM. When you install Warp, any existing versions of DOS and Windows, as well as all your applications, will be left untouched.

Warp features a graphical Workplace Shell (WPS). It's the essential "front door" to OS/2, and you'll need

to use it, even if you plan to run only DOS applications. (See the accompanying screen shot, page 50.) The first time you see WPS, it can be daunting and confusing. Although it's graphical, it doesn't look like Windows; in fact, it's similar in many ways to the Macintosh user interface. You'll need to polish your mouse skills, but extensive on-screen help is available to keep you from getting lost.

Juggling Tasks

Warp's biggest advantage is true multitasking, sometimes known as "preemptive" multitasking, letting you run multiple DOS applications at the same time. It differs from Windows' "cooperative" multitasking, where you can start multiple DOS sessions but operate only one at a time.

In Warp, so-called virtual DOS machines, or VDMs, act as an exact software emulation of DOS 5 operating under OS/2, so you can run many DOS sessions simultaneously without the utilities you'd normally find in DOS (or add on later). For example, Warp includes no SmartDrive disk-caching utility because

its own disk cache is more sophisticated, working with all applications—OS/2, Windows, and DOS.

When you want to run multiple DOS applications, Warp uses a sophisticated memory-switching technology, swapping applications to hard disk when it runs out of RAM. In theory, that means you can have up to 240 DOS sessions running at the same time.

You can run your DOS programs either in the familiar full-screen setup or in a window you can size to your preference and move around the screen. Switching among active applications is simple: Pressing Alt+Esc calls up Warp's Task Manager for a list of all running files. All you need to do is point to the one you want to bring to the front and press Enter.

If you like, you can even have each DOS application run its own

TOP TEN WARP TUNING TIPS

You can fine-tune each DOS session in OS/2 Warp, called a VDM (virtual DOS machine), as though it were running alone on a standard DOS PC. Warp offers more than 50 parameters for each DOS session, divided into seven categories. You'll never need to touch most of them, but some can make substantial differences both in the way individual DOS applications run and in overall OS/2 system performance. Here are some you should know about.

To access these settings, move the mouse pointer onto the icon for a DOS application, press the right mouse button, and highlight the Settings option to bring up the corresponding display. (See the first screen shot, below right.)

Then click on the Session tab and select the DOS Settings button to bring up a list of DOS-option categories. (See the second screen shot, page 51, top.) It sounds complex, but it'll soon become second nature.

1. COM_DIRECT_ACCESS (Other DOS settings)
COM_DIRECT_ACCESS gives DOS applications and drivers that use the serial port (such as communications programs and mouse drivers) direct access to your computer's serial ports instead of passing through OS/2's serial support layers. (See the third screen shot, page 51, bottom.) Setting COM_DIRECT_ACCESS to On can help improve performance and eliminate communications problems, especially with high-speed modems. Default is Off. (All communications go through OS/2 layers.)

2. DOS_AUTOEXEC (Other DOS settings)
DOS_AUTOEXEC specifies an alternate name for the AUTOEXEC.BAT file that runs when a particular VDM (DOS application) starts. This lets you specify an alternate batch file that runs instead of the standard DOS AUTOEXEC.BAT (which otherwise runs each time you start a DOS application within Warp). For example, let's say you use your mouse in only one of your DOS applications. Make a copy of your existing AUTOEXEC.BAT, rename it (USEMOUSE.BAT, for example), add the command for your mouse (usually MOUSE.COM), and from the DOS settings specify USEMOUSE-.BAT to run when you start this DOS session. The next time you start the DOS application, your system will ignore the DOS AUTOEXEC.BAT, and USEMOUSE.BAT will run instead. Default is DOS AUTOEXEC.BAT.
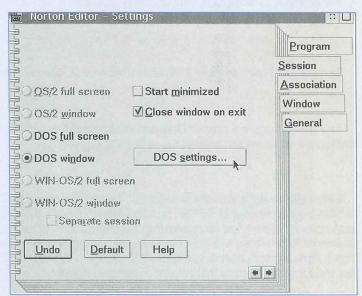
3. DOS_BACKGROUND_EXECUTION (Other DOS settings)
DOS_BACKGROUND_EXECUTION lets you choose whether DOS programs should continue running after you switch them to the background. You'll find that although certain programs, including communications software, should continue to run in the background, other applications, such as word processors, don't

To access Warp's DOS system-performance settings, move the mouse pointer onto the DOS app's icon and click; then click on Settings. You'll see the screen depicted above.
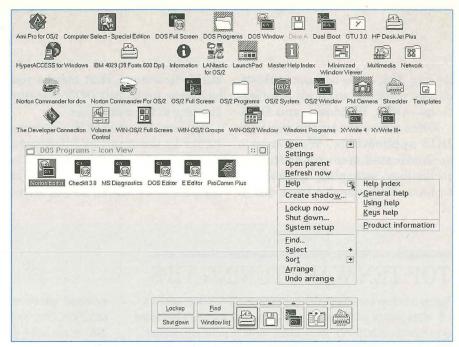
unique CONFIG.SYS and AUTOEXEC-.BAT files, letting you, for example, load a mouse driver only for DOS applications that need it, or load resource-hungry sound-card drivers only for games that use them.

Managing Memory

Unlike DOS, which features 640K of base memory, 384K of upper memory, and usually everything above the 1MB mark in extended memory, Warp uses *all* memory available in your PC.

Using all memory means that individual applications aren't bound by DOS's 640K limit—they can be large and sophisticated.

Running multiple DOS sessions, however, is a different story. As a group, because VDMs emulate DOS 5, they're still bound by that

Warp's graphical Workplace Shell (WPS): the "front door" to OS/2.

DOS TIP — SPEEDY SETUP

All of us have experienced the boredom of installing a new version of DOS. You read the installation screens, press buttons to confirm the default selections, and swap floppy disk after floppy disk. Although you'd never know it from reading Microsoft's documentation, the setup utility that comes with MS-DOS 6.x offers a shortcut or two.

For example, to install DOS without having to create the "uninstall" disks, use the "gone" switch:

SETUP /G

To install DOS using the default response for all choices, use the "hurry" switch:

SETUP /H

And to install only the optional DOS and Windows utilities, such as the backup and compression programs, use the "extra" switch:

SETUP /E

Before you go out and try these switches, however, check that they're available with your copy of SETUP, because certain versions of MS-DOS that come preinstalled on new computers may not offer all of the above choices. To see which switches are available, put your setup disk (disk 1) into your boot floppy drive, log onto drive A, and type SETUP /?. If you try this with PC DOS 6.x and 7.x, you'll discover that only the /E switch is available.

—Robert L. Hummel

640K "wall" that's been designed into DOS from the very beginning.
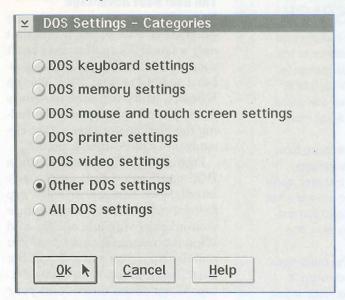
Although Warp's DOS emulation isn't perfect, you can run nearly all DOS applications—including certain programs, such as Microsoft Flight Simulator, that are finicky about the hardware on which they run—in virtual DOS machines without tweaking.

Other fussy applications, such as peer-to-peer networks like LANtastic, require special handling because they get confused by the layers of OS/2 underlying the DOS emulation. (An OS/2-specific version of LANtastic is available to combat this problem.)

For DOS applications requiring specific versions of DOS other than emulated 5, Warp offers a built-in workaround called VMDISK, which lets you create sessions that boot "true" versions of DOS. It's a multi-step process using original DOS disks (MS-DOS 6.22, for example), the details of which are beyond the scope of this article.

For a list of ideas for OS/2 customization, see the accompanying sidebar "Top Ten Warp Tuning Tips" (pages 49 and 51).

DOS Settings – Categories

- ○ DOS keyboard settings
- ○ DOS memory settings
- ○ DOS mouse and touch screen settings
- ○ DOS printer settings
- ○ DOS video settings
- ◉ Other DOS settings
- ○ All DOS settings

[Ok ▶]   [Cancel]   [Help]

You can fine-tune these categories of DOS settings in OS/2 Warp for better performance.

usually need to. If your DOS application doesn't need to run in the background, setting the DOS_BACKGROUND_EXECUTION option to Off can increase overall system performance, especially if you have a slower processor or limited RAM. Default is On. (All background applications keep running.)

4. DOS_DEVICE (Other DOS settings)

DOS_DEVICE creates a custom CONFIG.SYS for each DOS session, letting you load only the device drivers you need to run the DOS application assigned to the VDM. Entering just the driver name makes it exclusive to the VDM; entering DEVICE= in front of the driver name loads the driver into all DOS sessions. For example, if you want to use a sound card only in one particular DOS application, open the Settings menu and enter the name of the sound-card driver. On the other hand, if you want to use your sound card in all DOS sessions, open the settings for any DOS application and precede the sound-card driver name with DEVICE=. Default is no drivers loaded for DOS sessions.

5. DOS_HIGH (DOS memory settings)

Similar to DOS=HIGH, found in MS-DOS version 5 and later, this DOS_HIGH command has a crucial difference under OS/2. When on, it loads DOS into memory above 1MB (HMA, or high-memory area) instead of into the upper-memory blocks below 1MB, which MS-DOS uses. This design leaves additional upper-memory blocks free for loading device drivers or memory-resident programs. Default is Off (DOS not loaded in HMA).

6. DOS_UMB (DOS memory settings)

DOS_UMB lets you load device drivers and memory-resident programs into upper-memory blocks with LOAD-

HIGH and DEVICEHIGH in AUTOEXEC and DOS_DEVICE. Default is Off (no UMBs used, even if LOADHIGH or DEVICE-HIGH is specified).

7. EMS_MEMORY_LIMIT (DOS memory settings)

EMS_MEMORY_LIMIT sets the amount of EMS (expanded memory) available to the VDM. Some applications such as Microsoft Flight Simulator and Borland's database Paradox require it. OS/2's default setting is 2K of expanded memory. Setting it to zero saves memory and improves performance for applications that don't require expanded memory. Set it higher if your application uses additional extended memory (and the additional RAM is installed in your PC).

8. HW_ROM_TO_RAM (Other DOS settings)

The BIOS (basic input/output system), located in ROM (read-only memory), acts as an interface between your computer's specific hardware and the operating system. Copying the BIOS to RAM with the HW_ROM_TO_RAM option, instead of leaving it in relatively slow ROM, can increase performance with some applications. Default is Off (BIOS not copied to RAM).
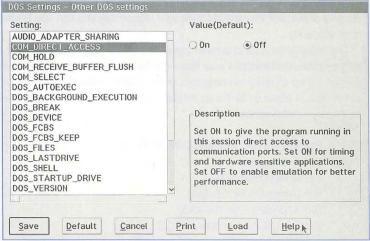
9. HW_TIMER (Other DOS settings)

HW_TIMER is one of the key settings for getting reluctant, "badly behaved" DOS applications (especially games) to run in VDMs. Turning this on gives applications direct access to your PC's hardware. Use it sparingly, because it can slow down overall system performance. Default is Off.

10. VIDEO_MODE_RESTRICTION (DOS video settings)

Character-mode DOS applications that don't use graphics can gain up to 736K of base memory when you turn on the VIDEO_MODE_RESTRICTION option. It lets the main application use the RAM usually reserved for DOS graphics. Default is Off.

—S.M.

DOS Settings – Other DOS settings

Setting:
- AUDIO_ADAPTER_SHARING
- COM_DIRECT_ACCESS
- COM_HOLD
- COM_RECEIVE_BUFFER_FLUSH
- COM_SELECT
- DOS_AUTOEXEC
- DOS_BACKGROUND_EXECUTION
- DOS_BREAK
- DOS_DEVICE
- DOS_FCBS
- DOS_FCBS_KEEP
- DOS_FILES
- DOS_LASTDRIVE
- DOS_SHELL
- DOS_STARTUP_DRIVE
- DOS_VERSION

Value(Default):
- ○ On   ◉ Off

Description
Set ON to give the program running in this session direct access to communication ports. Set ON for timing and hardware sensitive applications. Set OFF to enable emulation for better performance.

[Save] [Default] [Cancel] [Print] [Load] [Help ▶]

Setting COM_DIRECT_ACCESS to On can help improve high-speed modem performance and eliminate communications problems.

STACKER SQUEEZES SURPLUS SPACE

Despite the fact that contemporary PCs usually come with large-capacity hard drives and that large add-in drives for older systems are available for surprisingly low prices, you never, ever, have enough disk space—data just seems to expand to fill available space. The situation gets even trickier if you're an OS/2 user. Even a minimal Warp installation takes up about 35MB of hard-disk space. Add all the options, and you're up to about 60MB. And then there are all those space-hungry applications.

Stacker 4.0 for OS/2 and DOS offers a cutting-edge solution to wringing more space from your hard disk. Using a variety of methods, Stacker compression forces files into the minimum amount of space. The bottom line is that after you've installed Stacker, your hard disk will be twice as large, maybe even two and a half times larger. Your 200MB hard disk, for example, will hold between 400MB and 500MB of data. The exact size will depend on the types of files you store. You can't compress applications, for example, as much as text files.

Stacker is completely transparent and easy to install, although the usual caveat applies: Make sure you do a full backup of your hard disk before you begin. If you're using a dual boot, you can install Stacker from either DOS or OS/2. In fact, if you currently have a DOS-only system, you can install Stacker and have it ready to go if you decide to install OS/2 later. Installation is automatic if you want to compress all space on all your disks, but somewhat more complex if you want to manually "stack" only certain disks or parts of disks. Either way, once you install Stacker, there are no additional steps before you use it. And installation won't affect your existing hardware and software.

The original version of Stacker for OS/2 didn't work under DOS. For example, if you used OS/2's Dual Boot feature (the ability to start your PC under either DOS or OS/2) with the original version, you'd be out of luck if you started DOS, because you couldn't access Stacker-compressed files under DOS. But now you can in version 4.0.

The newest Stacker for OS/2 is also considerably faster—typically, three to four times—than its predecessor. In most cases, there's no perceptible slowdown when running Stacker-compressed files. And the faster your processor, the better the performance. It also includes several built-in safety features, in the unlikely event that problems arise with your compressed files. Even if you erase all your Stacker-compressed files, you can still recover them.

If you want, Stacker will let you fine-tune the ratio of speed to compression, trading off some compression for faster performance. This built-in utility is especially valuable if your PC has an older 386 processor. Other utilities will check the integrity of Stacker files and optimize your files so that you can use Stacker space most efficiently.

Trusting all your data to another layer of software—especially one that essentially encodes and decodes your files—can be scary. But I've used Stacker since version 1.0 on a variety of computers and hard disks without any problem. For a street price of about $100, it's an inexpensive and effective way to dramatically increase your disk space.

Stacker 4.0 for OS/2 and DOS, $100 retail
Stac Electronics
12636 High Bluff Drive
San Diego, CA 92130-2093
800-522-7822, 619-794-4300
requires OS/2 2.1 or later or DOS 5 or later

—S.M.

The Dual Boot Advantage

Even if you want to run only one program at a time, you'll find that only a few DOS applications won't cooperate with Warp. In the stubborn minority are The Norton Utilities, which accesses hardware directly, and some games (including the blockbuster Doom) that circumvent DOS/PC interaction.

Fortunately, if you need to run a DOS application that OS/2 won't handle, you can tap into a Warp feature called Dual Boot. Because your original versions of DOS and Windows remain intact when you install Warp, you can click on the Dual Boot icon from the OS/2 desktop; Warp shuts down, and your PC reboots from DOS, just as it did before you installed Warp.

Then, whenever you want, you can start Warp again from DOS simply by typing a command at the C:\ prompt.

Unless you run many DOS applications at once, you won't notice any significant slowdowns. Some programs, when run simultaneously with other DOS applications, may show a 3- to 5-percent speed decrease, but that's pretty insignificant.

Conversely, because of Warp's advanced caching and disk management, applications that do a lot of reading from and writing to your hard disk (databases, for example) will often run two to three times faster than under DOS.

New Dimension, New Power

While OS/2 Warp isn't perfect—no operating system is—it gets more from your hardware than DOS or Windows.

Sure, it takes time and patience to fully tap Warp's power, especially getting used to the Workplace Shell graphical interface and dealing with a confusing array of new options. But it's well worth the journey.

In a way, OS/2 Warp gives new life to DOS, wringing still more power and versatility from that venerable operating system. ■

QBASIC BRAIN TEASER

This Is Madness

Test your genius for solving puzzles with this computerized game of solitaire. It's easy to play—but so hard to win . . . .
Those little red and blue squares will keep you coming back for more.

by Dennis Mull

Nothing to it." Almost before the words were out of my mouth, I regretted them. More than a half hour later, I was still trying to solve a deceptively simple wooden puzzle my daughter had received as a present. It was maddening. When no one was looking, I tucked the game away, only to pick it up again after a few minutes. In the end, my perseverance paid off.

Still, even after I'd solved the puzzle, I couldn't get it off my mind. This time, though, I was pondering how to create an electronic version of the game. It wouldn't take long to write, I figured, because the game's rules are so simple. I knew just what to call it: MADNESS.BAS.

Play centers on a board consisting of a row of ten squares. Each of the four squares at the left end of the board contains a red playing piece; each of the four squares on the right end contains a blue piece. To win, you must move the red pieces to the right end of the board and the blue pieces to the left end. But you can move a piece only by

Dennis Mull is a programmer and is co-owner of D&T Soft Computer Service, a company specializing in installing and troubleshooting computer systems.

jumping over another piece or by advancing it one or two blank spaces at time. You can't move backward. Think you can figure it out? Have a try!

On Your Way

If you're anxious to put yourself to the test, you'll probably want to take the expedient approach to obtaining MADNESS.BAS: downloading the listing from the *DOS World* bulletin-board system. (See the "BBS" section of "How to Use This Magazine," page 64 in this issue, for assistance.) But if you're in no hurry and value the experience of learning programming by doing, why not type in MADNESS? (See the listing, beginning on page 55.)

To start, load QBasic by typing this command at the DOS prompt:

```
C:\DOS\QBASIC
```

Change the path if you store QBasic in a subdirectory other than C:\DOS. Press Esc to clear QBasic's opening screen, and immediately save the listing as MADNESS.BAS by choosing File/Save (or File/Save As) and typing its path and name in the File Name box. For example, if you want to store the file in C:\QBASIC \PROGRAMS, type this in the box:

```
C:\QBASIC\PROGRAMS\MADNESS.BAS
```

Next, begin typing the main program, starting with the DEFINT A-Z statement and continuing with the COMMON SHARED line and the lines that follow it. Skip the DECLARE SUB statements, because QBasic will add these lines automatically later on, when you create the program's subroutines.

As you work, try to remember to save your work every few minutes. When you finish typing the DATA statements and the LOOP statement that follows them, save the program again. You've reached the first subroutine, BOXFILL.

Type in the first line of BOXFILL (line 117, SUB BOXFILL . . .) exactly as shown. When you press Enter, QBasic will move you to another screen and display these lines:

```
DEFINT A-Z
SUB BOXFILL (ROWR%, COLS%, CLR%,
    FILL%) STATIC

END SUB
```
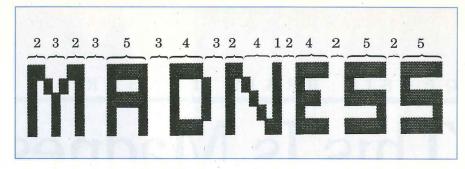
Position your cursor on the blank line, then type the five remaining lines of the subroutine. (Note that because the program begins with a DEFINT A-Z statement, QBasic adds this line to each subroutine.)

To begin the next subroutine, EXAMPLE, press F2 to bring up the Subs dialog box; highlight the main program, MADNESS.BAS; press Enter; and type SUB EXAMPLE.

After typing in the last line of this subroutine, save the listing again, then return to the main program as before.

Follow the same procedures for creating the remaining subroutines, double-checking your typing

This chart indicates the spacing of the special characters you use to create the MADNESS portion of the accompanying listing.

BEEP TIMES TWO

If you want to create a batch file that beeps, you have two choices. One approach is to ECHO the ASCII bell character to the screen with this command:

```
ECHO ^G
```

To produce the sequence ^G, hold down the Alt key while pressing the number 7 on the numeric keypad. (When you use TYPE to display the batch file on screen, ^G is invisible.)

But if you have a hard time remembering Alt-key combinations, you may want to try the other method: creating a short program called BEEP.COM. First, use DOS's EDIT.COM (or another editor that saves files in text format) to create a text file named BEEP.SCR. Remember to put a blank line between INT 21 and RCX. Next, type this command to create BEEP.COM:

```
DEBUG < BEEP.SCR
```

Then store the program in a directory listed in your AUTOEXEC.BAT's PATH command. Whenever you want to make your batch files beep, simply add this line:

```
BEEP
```

—Doug Lowe

Use the Debug script BEEP.SCR to create the executable program BEEP.COM, which you can use to sound a tone in your batch files.

```
A 100
MOV AH,2
MOV DL,07
INT 21
MOV AL,0
MOV AH,4C
INT 21

RCX
12
N BEEP.COM
W
Q
```

and fixing any errors that QBasic points out.

The only tricky part of typing in the program comes in the subroutine called MAKESCRN, which uses a series of extended ASCII characters to display the game's name on the playing screen.

To create the first line of code containing this special character, start by typing PRINT, then press the spacebar once, type a double quotation mark, and press the spacebar 13 times.

Next, create the special character by pressing the Num Lock key to activate the numeric keypad, then hold down the Alt key and press the numbers 1, 7, and 8 in succession.

The final display contains 109 of these characters, but, fortunately, you don't have to repeat the Alt-key sequence to create the remaining 108 characters. Instead, press Num Lock again to activate the cursor keys.

With the cursor on the character you just created, highlight it by pressing the Shift key and then the right-arrow key. To store the character in memory, hold down the Ctrl key and press the Insert key. With the character in memory, replicate it by placing the cursor in the correct spot and pressing Shift+Insert. (See the illustration above for help counting spaces and positioning the characters.)

Other Points of Interest

As you're typing, pay attention to several other features of the program: the subroutines KEYKILL,

End

MADNESS.BAS. Line numbers are for reference only; don't type them in. To create the special characters spelling out MADNESS, press the Num Lock key (on), then hold down the Alt key and type 178 on the numeric keypad. Also in that section note that you should press the spacebar 13 times between the initial quote mark and the first special character, although for printing purposes spaces appear narrower here. (See chart, opposite.)

```
1  'Dennis Mull
2
3  DEFINT A-Z
4  DECLARE SUB BOXFILL (ROWR%, COLS%, CLR%, FILL%)
5  DECLARE SUB EXAMPLE (HOWMANY%)
6  DECLARE SUB GAMEWIN ()
7  DECLARE SUB GETCOLUMN (COLUMN$)
8  DECLARE SUB KEYKILL ()
9  DECLARE SUB MAKESCRN (ROWSTART%, HITE%, WIDE%)
10 DECLARE SUB MAKESOUND (WHICH%)
11 DECLARE SUB WARNING (MESSAGE%)
12 DECLARE SUB WINCHECK ()
13
14 COMMON SHARED COUNT%, ATTEMPT%, COLS%, GEN$
15 DIM SHARED BOX$(11), KEEP$(30)
16
17 FOR REPLAY% = 1 TO 30      'Get example
      winning data
18   READ A$: KEEP$(REPLAY%) = A$
19 NEXT REPLAY%
20
21 GEN$ = "YOU ARE A GENIUS"
22 MAKESCRN 10, 5, 70
23 EXAMPLE REPLAY%              'Play example game
24 TOP:
25 COUNT% = 0: ATTEMPT% = ATTEMPT% + 1
26 IF ATTEMPT% > 4 THEN GEN$ = "Sorry, genius
      status denied"
27 MAKESCRN 10, 5, 70
28
29 DO
30 REDO:
31   DO
32   KEYKILL      'Remove keystrokes from keyboard
33   LOCATE 9, 14: PRINT "Number of Moves > ";
        COUNT%
34   LOCATE 9, 44: PRINT "Number of Attempts > ";
        ATTEMPT%
35   LOCATE 20, 31: PRINT "    ": LOCATE 20, 69:
        PRINT "    "
36
37   LOCATE 20, 30: INPUT " ", A$
38   IF UCASE$(A$) = "Q" THEN COLOR 7, 0: CLS :
        END    'Quit MADNESS
39   IF UCASE$(A$) = "N" THEN GOTO TOP
        'Start new game
40
41   LOCATE 20, 68: INPUT " ", B$
42   AA% = VAL(A$): BB% = VAL(B$)
43   IF AA% = 0 OR BB% = 0 THEN MAKESOUND 2:
        GOTO REDO
44   IF AA% > 10 OR BB% > 10 THEN AA% = 1:
        BB% = AA%
45   IF AA% < 1 OR BB% < 1 THEN AA% = 1: BB% = AA%
46
47   IF BOX$(BB%) <> "BLANK" THEN WARNING 3: A$ = ""
48
49   IF A$ <> "" THEN
50     IF BOX$(AA%) = "BLANK" THEN WARNING 4: A$ = ""
51   END IF
52
53   IF A$ <> "" THEN
54     IF AA% > BB% THEN        'Check for jump > 1
55       IF AA% - BB% > 2 THEN WARNING 1: A$ = ""
56     END IF
57   END IF
58
59   IF AA% < BB% THEN          'Check for jump > 1
60     IF BB% - AA% > 2 THEN WARNING 1: A$ = ""
61   END IF
62
63   IF A$ <> "" THEN
64     IF BOX$(AA%) = "RED" THEN   'Check for
          backward move by red
65       IF BB% > AA% THEN
66       EXIT DO
67       ELSE
68       MAKESOUND 2: WARNING 2: A$ = ""
69       END IF
70     END IF
71   END IF
72
73   IF A$ <> "" THEN
74     IF BOX$(AA%) = "BLUE" THEN  'Check for
          backward move by blue
75       IF BB% < AA% THEN
76       EXIT DO
77       ELSE
78       MAKESOUND 2: WARNING 2: A$ = ""
79       END IF
80     END IF
81   END IF
82 LOOP WHILE A$ = "" OR B$ = ""
83
84 COUNT% = COUNT% + 1          'Count the moves
85 LOCATE 9, 14: PRINT "Number of Moves > ";
      COUNT%
86 LOCATE 9, 44: PRINT "Number of Attempts > ";
      ATTEMPT%
87
88 GETCOLUMN A$
89
90 IF BOX$(VAL(B$)) = "BLANK" THEN 'Change to
      blank square
91   FILL% = 32
92   BOXFILL 11, COLS%, CLR%, FILL%
93   BOX$(VAL(B$)) = BOX$(VAL(A$))
94 END IF
95
96 GETCOLUMN B$
97
98 IF BOX$(VAL(A$)) = "RED" THEN    'Check for
      color swap
99   CLR% = 4
100 ELSE CLR% = 1
101 END IF
102
103 BOX$(VAL(A$)) = "BLANK"
104 FILL% = 186      'Change moved box character
105 BOXFILL 11, COLS%, CLR%, FILL%
106 MAKESOUND 5
107 WINCHECK                 'Check for winning game
```

```
108  COLOR 15, Ø
109  KEEP$(COUNT%) = STR$(AA%) + STR$(BB%)
        'Keep track of moves
11Ø
111 'Example data of winning moves
112 DATA " 7 5"," 8 7"," 4 6"," 3 4"," 6 8";"
        7 6"," 5 3"," 9 7"," 8 9"," 6 5"
113 DATA " 4 6"," 2 4"," 1 2"," 3 1"," 5 3","
        1Ø 8"," 7 5"," 9 1Ø"," 8 7"," 6 8"
114 DATA " 8 9"," 4 6"," 6 8"," 2 4"," 4 6","
        3 2"," 5 3"," 7 5"," 5 4"," 6 7"
115 LOOP
116
117 SUB BOXFILL (ROWR%, COLS%, CLR%, FILL%) STATIC
118  F$ = STRING$(6, FILL%)      'Define contents
        of box
119  FOR R = 11 TO ROWR% + 2
12Ø   COLOR CLR%, Ø: LOCATE R, COLS%: PRINT F$
121  NEXT
122  COLOR 15, Ø
123 END SUB
124
125 SUB EXAMPLE (HOWMANY%) STATIC
126
127  FOR REP% = 1 TO HOWMANY% - 1
128   TIN! = TIMER: WHILE TIMER < (TIN! + .1): WEND
129   A$ = LEFT$(KEEP$(REP%), 3): A$ =
        LTRIM$(RTRIM$(A$))
13Ø   B$ = RIGHT$(KEEP$(REP%), 2): B$ =
        LTRIM$(RTRIM$(B$))
131   LOCATE 9, 14: PRINT "Number of Moves > "; REP%
132   LOCATE 9, 44: PRINT "Number of Attempts > ";
        ATTEMPT%
133
134   GETCOLUMN A$
135   IF BOX$(VAL(B$)) = "BLANK" THEN
136    FILL% = 32
137    BOXFILL 11, COLS%, CLR%, FILL%
138    BOX$(VAL(B$)) = BOX$(VAL(A$))
139   END IF
14Ø   GETCOLUMN B$
141   IF BOX$(VAL(A$)) = "RED" THEN
142    CLR% = 4
143   ELSE CLR% = 1
144   END IF
145
146   BOX$(VAL(A$)) = "BLANK": FILL% = 186
147   BOXFILL 11, COLS%, CLR%, FILL%
148   WINCHECK
149   MAKESOUND 4
15Ø  NEXT REP%
151  SLEEP 3: COLOR 15, Ø
152
153 END SUB
154
155 SUB GETCOLUMN (COLUMN$) STATIC
156  SELECT CASE COLUMN$      'Set column
        position of square
157   CASE "1": COLS% = 6:   CASE "2":  COLS% = 13
158   CASE "3": COLS% = 2Ø:  CASE "4":  COLS% = 27
159   CASE "5": COLS% = 34:  CASE "6":  COLS% = 41
16Ø   CASE "7": COLS% = 48:  CASE "8":  COLS% = 55
161   CASE "9": COLS% = 62:  CASE "1Ø": COLS% = 69
```

```
162   END SELECT
163 END SUB
164
165 DEFSNG A-Z
166 SUB KEYKILL STATIC
167  WHILE INKEY$ <> "": WEND
168 END SUB
169
170 SUB MAKESCRN (ROWSTART%, HITE%, WIDE%) STATIC
171  CLS : N% = Ø
172  LOCATE 2
173  COLOR 7, Ø
174  PRINT "███ ██ ███ ██ █ ███ ██ █"
175  COLOR 5, Ø
176  PRINT "████ █ █ █ █ ███ ██ █   █  "
177  COLOR 4, Ø
178  PRINT "███ █ ███ █ ███ ███ ███ █"
179  COLOR 3, Ø
180  PRINT "███ █ ███ █ ███ ███   █  █"
181  COLOR 2, Ø
182  PRINT "███ █ █ █ █ ███ ██ ███ ███"
183  COLOR 15, Ø
185
185  WINBOX% = ROWSTART%
186  COL% = (8Ø - WIDE%) / 2
187  P% = 2Ø1: T% = 2Ø5: Q% = 187        'Box
        characters
188  U% = 186: R% = 2ØØ: S% = 188        'for
        double-line box.
189  LOCATE WINBOX%, COL%, Ø
19Ø  PRINT CHR$(P%); STRING$(WIDE% - 1, T%);
        CHR$(Q%)  'First line box
191  FOR A% = 1 TO HITE% - 2
192   LOCATE (WINBOX% + A%), COL%, Ø
193   PRINT CHR$(U%); SPC(WIDE% - 1); CHR$(U%)
        'Body of box
194  NEXT
195  LOCATE WINBOX% + A%, COL%, Ø
196  PRINT CHR$(R%); STRING$(WIDE% - 1, T%);
        CHR$(S%); 'Bottom line box
197
198  FOR T = 12 TO 72 STEP 7        'Build rest
        of game board
199   LOCATE 1Ø, T: PRINT CHR$(2Ø3)
2ØØ   LOCATE 11, T: PRINT CHR$(186)
2Ø1   LOCATE 12, T: PRINT CHR$(186)
2Ø2   LOCATE 13, T: PRINT CHR$(186)
2Ø3   LOCATE 14, T: PRINT CHR$(2Ø2)
2Ø4   N% = N% + 1
2Ø5   LOCATE 16, T - 5: PRINT N%
2Ø6  NEXT T
2Ø7   LOCATE 16, T - 4: PRINT "1Ø"
2Ø8
2Ø9  FOR COLS% = 6 TO 27 STEP 7  'Set red squares
21Ø   BOXFILL 11, COLS%, 4, 178
211  NEXT
212
213  FOR COLS% = 48 TO 73 STEP 7   'Set blue squares
214   BOXFILL 11, COLS%, 1, 178
215  NEXT
216
217  FOR X = 1 TO 4: BOX$(X) = "RED": NEXT
218   BOX$(5) = "BLANK"
219   BOX$(6) = "BLANK"
22Ø  FOR X = 7 TO 1Ø: BOX$(X) = "BLUE": NEXT
221
```

```
222   LOCATE 20, 10: PRINT "MOVE SQUARE NUMBER >"
223   LOCATE 20, 50: PRINT "TO SQUARE NUMBER >"
224   LOCATE 9, 14: PRINT "Number of Moves > ";
         COUNT%
225   LOCATE 9, 44: PRINT "Number of Attempts > ";
         ATTEMPT%
226   COLOR 15, 6
227   LOCATE 24, 1: PRINT "  Q=Quit  N=New Game";
         SPACE$(58);
228   COLOR 15, 0
229 END SUB
230
231 SUB MAKESOUND (WHICH%) STATIC    'Sound generator
232   SELECT CASE WHICH%
233   CASE 1: SOUND 1800, 5
234   CASE 2: SOUND 50, 5
235   CASE 3
236     FOR X = 40 TO 1 STEP -1       'Make sound
            for winning game
237       SOUND 1000 - X * 10, .5
238       SOUND 100 + X * 10, .5
239     NEXT X
240   CASE 4: SOUND 1100, 1
241   CASE 5
242     FOR X = 5 TO 1 STEP -1        'Make sound
            for moving a piece
243       SOUND 1000 - X * 10, .6
244       SOUND 100 + X * 10, .2
245     NEXT X
246   END SELECT
247 END SUB
248
249 SUB WARNING (MESSAGE) STATIC
250
251   SELECT CASE MESSAGE
252   CASE 1: MESS$ = "  You may jump only one
            square at a time"
253   CASE 2: MESS$ = "   Sorry, you may not
            jump backward"
254   CASE 3: MESS$ = "Sorry, you must move to
            an empty square"
255   CASE 4: MESS$ = "  Sorry, you may not move
            an empty square"
256   END SELECT
257
258   MAKESOUND 2: LOCATE 22, 22: PRINT MESS$;
259   TIN! = TIMER: WHILE TIMER < (TIN! + 2): WEND
260   LOCATE 22, 22: PRINT STRING$(40, 32);
261   COUNT% = COUNT% + 2
262   KEYKILL
263 END SUB
264
265 SUB WINCHECK STATIC
266
267   FOR WIN% = 1 TO 4
268     IF BOX$(WIN%) <> "BLUE" THEN YES% = 0:
            EXIT SUB
269     YES% = YES% + 1
270   NEXT WIN%
271
272   FOR WIN% = 7 TO 10
273     IF BOX$(WIN%) <> "RED" THEN YES% = 0:
            EXIT SUB
274     YES% = YES% + 1
275   NEXT WIN%
276
277   FOR WIN% = 5 TO 6
278     IF BOX$(WIN%) <> "BLANK" THEN YES% = 0:
            EXIT SUB
279     YES% = YES% + 1
280   NEXT WIN%
281
282   IF YES% = 10 THEN
283     ATTEMPT% = 0: COLOR 31, 6: LOCATE 24, 50:
            PRINT GEN$;
284     MAKESOUND 3
285   END IF
286   COLOR 15, 0
287 END SUB
```

End

MAKESOUND, and WARNING, plus the lines in the main program associated with them. You may find these routines useful in your own programs.

KEYKILL keeps unwanted keystrokes from being executed while the program is handling other tasks. To achieve this, it clears the keyboard buffer before another INPUT statement is executed.

MAKESOUND provides control over the game's sound effects. I increased the versatility of this routine by designing it so that I could create new sounds merely by changing the value of the variable WHICH% and adding a CASE and a SOUND state-ment. When a particular CASE condition is met, the program plays the sound immediately after that CASE command. To play the sound, use a line such as this one:

```
IF AA% = 0 OR BB% = 0 THEN
   MAKESOUND 2: GOTO REDO
```

If the value of AA% or BB% is zero, the program executes the MAKE-SOUND subroutine and sets the value of WHICH% to 2. As a result, the CASE 2 sound—SOUND 50, 5—plays.

The structure of WARNING is comparable to that of MAKESOUND. This time, however, instead of trigger-ing different sounds, various values of WARNING trigger different messages.

As with MAKESOUND, you may add elements by appending CASE statements and their corresponding messages. Here, for instance, the value of WARNING is set to 4:

```
IF A$ <> "" THEN
   IF BOX$(AA%) = "BLANK" THEN
      WARNING 4: A$ = ""
END IF
```

When the second line executes, the message corresponding to CASE 4 is displayed: "Sorry, you may not move an empty square."
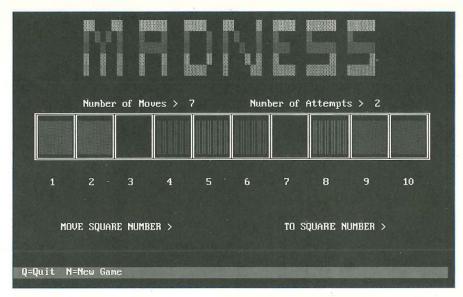
Game Time

With a working version of MADNESS.BAS in hand, you're ready to play. If QBasic and MADNESS.BAS are loaded, simply press Shift+F5 to run the program. If neither is loaded, type this command at the DOS prompt:

```
QBASIC /RUN MADNESS.BAS
```

Supply a path for MADNESS.BAS if it isn't in the current directory or in a directory mentioned in your PATH statement. The program will display a screen like the one in the accompanying screen shot, "MADNESS in progress" (right). Watch carefully; you might learn something, because the program begins

Madness in progress. MADNESS.BAS (which runs in full color on your screen) challenges you to move all the red game pieces to the right side of the board and all the blue pieces to the left. If you can figure out how to do it in four or fewer games, the program accords you genius status.

TALKING TO YOUR MODEM

Installing internal modems can be frustrating. If the modem doesn't work after you install it and try dialing out with your communications package, you don't know whether the hardware, software, or the communications (COM) port setting is at fault. Here's a quick way to check the COM-port assignment of a Hayes-compatible modem.

On most modems, the device's pins or switches determine the COM port to which the modem is set. If you're lucky, the product's manual will tell you which settings to use for each of your computer's COM ports, which are numbered from 1 to 4. Set the pins to the correct positions for the desired port. If you don't know which port to use, try COM2. Often a mouse is set up to use COM1, and the two devices can't share the port.

After you set the COM port, test the hardware by typing this command at the DOS prompt:

```
ECHO ATDTphone>COMx
```

where *phone* is an acceptable phone number and *x* is the port number. (This is only a test; no communication will take place, provided you hang up quickly enough.) For example, if you want to use COM2 to call 555-1234, type this command:

```
ECHO ATDT555-1234>COM2
```

If you hear a dial tone, the modem and the COM port are working, and you're ready to test your communications package. To hang up, type:

```
ECHO ATH>COM2
```

If you don't hear a dial tone, you've chosen the wrong COM port or set the modem's pins incorrectly. Change the settings and repeat the first ECHO command above until you hear the dial tone.

—Dan Keen

by providing a demonstration of one way to win the game.

After the demo ends, MADNESS.BAS resets the game board. Messages above the playing grid tell you the number of moves and the number of attempts in the current round of play. Below the grid are numbers identifying each square and prompts asking you to type the number of the piece you want to move and its new location. A highlighted bar at the bottom of the screen tells you to press the Q key to quit or the N key to start the game over.

Now comes the challenge: figuring out a sequence of jumps and advances that swap the positions of the red and blue pieces. Don't even think about trying to move a playing piece backward. If you do, the program sounds a buzzer, displays a "Sorry, you may not jump backward" message, and adds two extra moves to your score.

The good news is that if you meet success in four or fewer games, the program awards you genius status. If not, no harm done, and play continues until you decide to quit.

And I do advise you to quit every once in a while. After all, it's just a game. Or is it madness? ∎

WINDOWS SEAT

The Best Of Both Worlds

by Tony Roberts

Computing in both the DOS world and the Windows world simultaneously presents a challenge. Perhaps the biggest obstacle facing the DOS/Windows user is memory allocation—specifically, optimizing memory to run DOS software under Windows.

In most cases, a DOS/Windows user wants to maximize the amount of conventional memory available in DOS sessions running under Windows. If you do that, you can use Windows as a home base while launching and running DOS applications successfully.

> *The key to running resource-hungry DOS applications under Windows is to make sure enough conventional memory is available before Windows boots up.*

Conventional memory is that 640K of space into which most software running under DOS must fit. Under the standard DOS COMMAND.COM scenario, multitasking isn't an option. The active DOS program assumes it's the only game in town, so it takes all available conventional memory for its use. In contrast, once Windows is up and running, it carves out sections of extended memory as needed to run its various applications.

So what happens if you start two or more DOS applications from Windows? Which one gets access to conventional memory? They all do—or at least they all get tricked into thinking they do. When running in 386 Enhanced mode, Windows creates a "virtual machine" for each DOS session.

Each of these DOS sessions inherits the characteristics of the original DOS session—the one that launched Windows. So the key to running DOS applications that require significant chunks of conventional memory under Windows is to make sure that a maximum amount of this memory is available before Windows boots up.

Dedicated DOS users are no doubt experts in this area already, having spent years learning to get their applications and TSR (terminate and stay resident) programs to coexist. You've taken the steps allowed by recent DOS versions to load DOS itself high and to make use of upper-memory blocks, all of which provide more conventional memory.

From this starting point, you may want to take a few extra steps, as outlined here.

Mousing Around with DOS

When Windows loads, it activates its own mouse support. If you use your mouse only with Windows and Windows software, you don't need to run a mouse driver from your AUTOEXEC.BAT file. That can save you 15K or more.
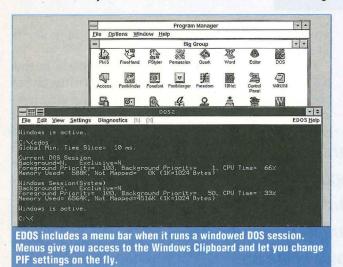
EDOS: ENHANCED DOS FOR WINDOWS

If you really want to take your DOS computing under Windows to another level, check out EDOS from Firefly Software Systems. This shareware program's most noticeable feature is that it lets you run DOS sessions with as much as 704K or 736K of memory available. These large DOS sessions won't run with graphics applications, but they do provide some additional elbow room for certain applications.

The extra memory is just the beginning. EDOS's command-line interface is actually a Windows virtual device driver. As such, it operates at the core of the Windows operating system, where it has access to any memory location, to the entire base of Windows code, and to all available hardware. The result is that you can run DOS sessions that can do things that DOS sessions normally can't. You can run DOS software while taking advantage of some of Windows' more convenient features, such as the Clipboard.

DOS sessions running in a window under EDOS include a menu bar, which gives you access to several features, including the Clipboard, file launching, and session settings. (See the screen shot, below left.) The latter lets you change PIF settings on the fly.

If you enjoy tuning your system for best possible performance, EDOS includes a few commands that will help. BOXTIME initiates a timer that lets you see how much CPU time your DOS session is getting. Use this information to track the effectiveness of your changes to various PIF settings. To get the scoop on what's going on with your machine, try the STATUS command; it provides information about memory use by both Windows and your DOS session, plus current system settings, swap-file details, and stack data.

EDOS lets you start Windows programs right from the DOS session command line, and it includes an alarm that's set in DOS but goes off in Windows. Using this feature, for example, you could start a batch file in a DOS session and then switch back to Windows, leaving the batch file working in the background. When the ALARM command (the last command in the batch file) is executed, a message pops up in Windows to let you know that your task has been completed.

If you have one foot firmly planted in DOS and the other in Windows, you'll welcome the way EDOS bridges the two systems. To order EDOS or for more information, contact Firefly Software Systems, P.O. Box 5035, Oregon City, OR 97045; 503-694-2282. Registration fee is $69.95.

—T.R.

EDOS includes a menu bar when it runs a windowed DOS session. Menus give you access to the Windows Clipboard and let you change PIF settings on the fly.

On the other hand, if any of your DOS applications requires the mouse, you'll have to load the mouse driver before starting Windows.

The MemMaker Challenge

MemMaker, the automatic memory-configuration utility that's part of DOS 6.x, can help you organize your drivers and TSR software to your best advantage. Third-party memory managers—such as QEMM—offer similar auto-configuration utilities. These programs help you maximize conventional memory by finding the most efficient way to fit your drivers and TSRs into upper memory.

If you use MemMaker's Custom Setup, you'll want to consider carefully how you answer the "Optimize Upper Memory for Use with Windows?" question, which appears on the Advanced Options page.

If you use DOS programs with Windows, you might want to answer yes to this question, but maybe not. If you answer yes, MemMaker attempts to optimize memory so that you have the maximum amount of conventional memory available when running DOS sessions from Windows. Although that sounds ideal, the benefit comes at a price: You may actually lose a good chunk of conventional memory when you're working from DOS and Windows isn't active.

To arrive at the best solution for the way you work, set aside some time and experiment with both options. Use the DOS MEM command to find out how much conventional memory is free before and after Windows is running, and test your software to see whether there's enough elbow room for it under each setup.

Load TSRs with Care

Take an inventory of the TSR programs you're running. You probably don't need to activate all of them all the time. For example, what good is DOSKEY when you're playing one of those memory-hungry games?

One way to boost conventional memory is to leave your TSR programs out of your startup routine and start them only for the specific DOS sessions that need them. You do this by writing a simple batch file that loads the TSRs and then the software. Once you've created the batch file, use PIFEDIT in Windows to create a PIF (program-information file) that executes your batch file. (See the first screen shot, top right, showing the PIF I use to run GEnie's Aladdin communications software.)
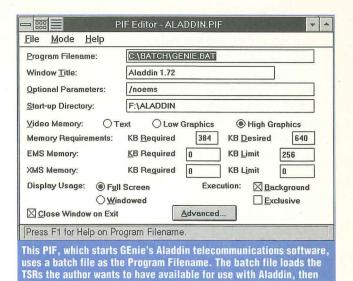
If you want to have certain TSRs running during a session at the DOS prompt, again, create a batch file that loads the TSRs, and specify this batch file as an optional parameter in your regular DOSPRMPT.PIF. (See the second screen shot, bottom right.) Note the KB Required and KB Desired boxes. Here I've specified –1 in each box: the functional equivalent of specifying 0 (zero) for KB Required and 640 for KB Desired. This shorthand tells Windows to allocate as much memory as possible—up to 640K—to the DOS session. Assuming Windows has enough free memory left in its memory pool, this setting ensures that you can run nearly any program, large or small, from the DOS prompt.

Some of you might be tempted to combine a couple of the previous suggestions and start your mouse driver from a batch file when running applications that need it. Good idea, but Windows won't cooperate. Windows knows that the mouse driver is operating in one place, and it won't let you load it again.

Don't Duplicate Drivers

An obvious way to keep memory clear is to avoid loading unnecessary drivers. This suggestion seemed almost too simple to mention until I recently discovered I was loading 36K of unnecessary software associated with my sound card.

Installing my sound card added drivers to my CONFIG.SYS file, so I figured they were necessary. When I subsequently ran out of RAM, I tried "REM-ming out" those drivers. Nothing happened; the sound card worked fine. Upon investigation, I learned that the files were DOS drivers used by the

This PIF, which starts GEnie's Aladdin telecommunications software, uses a batch file as the Program Filename. The batch file loads the TSRs the author wants to have available for use with Aladdin, then loads the Aladdin software.

This PIF shows how to use the /K switch to execute a batch file in conjunction with COMMAND.COM. The TSR.BAT file listed here includes the TSR programs, such as DOSKEY, that the author wants active during a DOS session. The –1s in the KB Required and KB Desired boxes indicate that he wants Windows to provide as much memory as possible to the DOS session.

PLAY and RECORD utilities that came with the sound card. Because I never used those programs, loading the drivers was pointless—and a waste of memory.

Don't Be a Quitter

It's frustrating to have to exit Windows to run a DOS program. What's more, it's unnecessary. With just a few tweaks to your system setup, you'll find that Windows can serve as a very capable DOS command center. ■

Tony Roberts owns and operates a desktop-publishing business and has been writing about personal computers for 12 years.

DOS TIP
BOOSTING YOUR COMMUNICATIONS SPEED

Most newer modems compress data, which means that to use them most efficiently, you must send them more data than their ratings indicate that they can transmit. For example, if you have a 9600-bps modem or a 14,400-bps modem, set your communication software to 19,200 bps. If you have a 2400-bps modem with V.42bis or MNP5 data compression, set your software to 9600 bps.

—Randy Steele

THE CONFIG.SYS FILE

In your root directory is a file called CONFIG.SYS. Like AUTOEXEC.BAT, this file is in ASCII, and you can view your CONFIG.SYS file with the TYPE command. A typical CONFIG.SYS might look like this:

```
DEVICE=C:\DOS\HIMEM.SYS
DEVICE=C:\DOS\EMM386.EXE NOEMS
DOS=HIGH,UMB
FILES=5Ø
BUFFERS=1Ø
SHELL=C:\DOS\COMMAND.COM
    C:\DOS\ /E:1Ø24 /P
DEVICE=C:\DOS\ANSI.SYS
DEVICE=C:\DOS\SETVER.EXE
```

The rules for handling CONFIG.SYS are the same as they are for AUTOEXEC.BAT: Always back up the original file before you modify it and always have an emergency boot disk available. As with AUTOEXEC.BAT, changes you make to CONFIG.SYS won't take effect until you restart your computer.

ANSI.SYS AND THE ESCAPE CHARACTER

When an article says you must have ANSI.SYS installed, it means that the MS-DOS file ANSI.SYS should be in your \DOS directory, and the following line should be in your CONFIG.SYS file:

`DEVICE=C:\DOS\ANSI.SYS`

Some articles that discuss ANSI.SYS will also ask you to create a batch file that uses the escape character. Unfortunately, there's no uniform method of doing so. If you use EDIT, the text editor that comes with MS-DOS, you can make an escape character by pressing Ctrl+P and then the Esc key. The escape character appears on screen as a small left-pointing arrow. If you're using another text editor or word processor, check its instructions for information on how to enter the escape character.

THE AUTOEXEC. BAT FILE

Most people have a batch file called AUTOEXEC.BAT on their hard disk. If you want to look at it, first go to your root directory by typing CD\. Type DIR to make sure AUTOEXEC.BAT is there. Then type the following command:

```
TYPE AUTOEXEC.BAT | MORE
```

A simple AUTOEXEC.BAT file might look like this:

```
@ECHO OFF
PROMPT $P$G
PATH=C:\DOS;C:\WINDOWS;C:\WP51;C:\BAT
C:\DOS\SMARTDRV.EXE
C:\MOUSE\MOUSE.COM
C:\DOS\DOSKEY.COM
SET TEMP=C:\TEMP
```

When a *DOS World* article instructs you to modify your AUTOEXEC.BAT file, always make a backup copy of the original AUTOEXEC.BAT first. The most common names for your backup copy are AUTOEXEC.BAK or AUTOEXEC.BK. The latter lets you save different versions of your backups—for example, AUTOEXEC.BK1 and AUTOEXEC.BK2. You create a backup copy with the following command:

```
COPY AUTOEXEC.BAT AUTOEXEC.BAK
```

Also, you should have an emergency boot disk available whenever you modify AUTOEXEC.BAT. (See the accompanying section on the facing page, top.) It will let you access your hard drive in case you make an error that locks up your computer. Changes you make to AUTOEXEC.BAT won't take effect until you restart your computer.

BATCH FILES

A batch file is a text file that tells MS-DOS to do a series of tasks. The filename of a batch file always ends with the extension .BAT.

A batch file must be in plain-text format. For example, a batch file might consist of the following lines:

```
CD\
DIR /S /P
```

This batch file moves you to the root directory (CD\) and then gives you a list of all files in all directories (/S), pausing after each full screen (/P).

Every batch file needs a name. In such cases, you should pick your own name. Batch-file names carry the same limitations as any other DOS filename; you're limited to eight characters, plus a three-character extension. A batch-file name must always use the .BAT extension.

To avoid confusion and unexpected results, don't give any batch file the same name as another program or DOS command. For example, VCOPY.BAT is an acceptable name for a batch file, but not COPY.BAT or XCOPY.BAT, because COPY and XCOPY are the names of DOS commands. To run or execute a batch file, type its name at the DOS prompt. For example, to run a batch file called VCOPY.BAT, type VCOPY at the DOS prompt.

Creating and Saving

Using EDIT. If you have DOS 5 or later, you can create a batch file using EDIT. EDIT usually resides in your DOS directory. Type EDIT and enter your batch file. When you're done, press Alt+F and choose the Save option. Type the name of your batch file (make sure you add the extension .BAT) and press the Enter key.

Using other word processors. Most word processors don't save files in plain text; they include other characters, such as control characters that handle such matters as page formatting and typefaces. Most word processors, however, do give you an option to save in plain text. The procedure varies from one word processor to the next. For example, when you save a file in WordPerfect 5.1, you choose ASCII Text (DOS) as your Format option.

QBASIC PROGRAMS

QBasic is the programming language included in all versions of MS-DOS since version 5. The name of a QBasic program always ends with the extension .BAS.

Typing in the listing. Type QBASIC at the DOS prompt and press Enter to start. Now type in the listing as printed, pressing Enter at the end of each line. Note that when a line in the listing is indented two spaces from the line above and doesn't start with a command or keyword, it's a continuation of the previous line. Other indentations, or none at all, indicate a new line.

Subroutines and functions. QBasic listings often include subroutines and functions, and typing them is confusing at first. They begin with a line containing the keyword SUB or FUNCTION. Note that when you type a SUB or FUNCTION line and press Enter, all other lines you've typed will disappear from view. This can be disconcerting for beginning programmers. There's nothing to worry about—your listing is safe. To avoid screen clutter, QBasic simply hides other parts of your listing when you're typing in a subroutine or function. To see the other parts of your program, open the View menu at the top of the QBasic screen, then select SUBS. The SUBS dialog box will appear, letting you select the part of the program listing you want to view.

Saving a listing. Save your partially completed listing as you go along, rather than waiting until you've typed in the whole thing. To save, open the File menu, choose Save, and type in a filename when QBasic prompts you. We suggest using the filename specified in the magazine article. Subsequent saves of your listing won't prompt you for a filename, but will instead use the filename indicated the last time you saved the listing.

Running a program. After you've typed in the entire listing and saved it a final time, you can run the program by selecting Start from the Run menu or pressing Shift+F5. If QBasic finds an error, it will stop the program and highlight that line. To run a QBasic program (a .BAS file) stored on your hard drive, start QBasic, then select Open from the File menu. Choose from among the .BAS files displayed in the open dialog box to load the program into QBasic, then select Start from the Run menu or press Shift+F5. To stop a QBasic program, press Ctrl+Break; select Exit from the File menu to return to DOS.

DOS World BBS. Typing and debugging a long listing is time-consuming. If you have a modem, our listings are always available on *DOS World*'s bulletin-board system (BBS) at 603-924-3181. There are no connect-time charges; you pay only for the phone call. Set your communication program to 8 data bits, no parity, 1 stop bit (8, N, 1). Dial the number and wait for the "Connect" message. If you're a first-time user, the system will ask you to enter your name and choose a password. Then it will display a general information screen, followed by a questionnaire requesting your address, phone number, and so on, so that we may set up your account. From this point, on-screen prompts are the same for all users. A series of messages present the latest BBS news; press Enter after each message to go to the next screen. From the Bulletin Menu, Bulletin #1 offers information on navigating the Main and File Menus, with instructions for listing, marking, searching for, and downloading files. ■

This Magazine

MAKING AN EMERGENCY BOOT DISK

Sometimes a *DOS World* article will suggest that you create a *bootable floppy*—a floppy disk that serves as an emergency system disk. That is, if your computer for some reason can't access your hard drive, you can start your computer from the emergency floppy. You should always have an emergency system disk available, but it's particularly important when you modify AUTOEXEC.BAT or CONFIG.SYS because you may change those files in such a way that your computer won't start from the hard drive. To create a system disk:

1. Insert a floppy disk in drive A.
2. At the command line, type FORMAT A: /S (all existing information on the floppy will be lost).

DOS first formats the floppy disk. Then it copies three DOS system files to the floppy disk: IO.SYS, MSDOS.SYS, and COMMAND.COM.

The first two are hidden files; you won't see them if you type DIR A:. If you have the disk-compression program Double-Space on your computer, the FORMAT command above will also copy DBL-SPACE.BIN, a third hidden file, to the floppy disk.

After you've created your system disk, you should copy a few other basic files to your floppy. Go to your \DOS directory and copy the following files: FORMAT-.COM, EDIT.COM, EDIT.HLP, QBASIC-.EXE, UNDELETE.EXE, CHKDSK.EXE, FDISK.EXE, and SETUP.EXE.

DEBUG SCRIPTS

A Debug script is a list of assembly-language instructions you convert to an executable program using the program DEBUG.EXE in your \DOS directory.

Creating the script. A Debug script must be in plain text. The procedure for creating the script is the same as for creating a batch file. You can use DOS's EDIT program, or you can use a different text editor or word processor and save the script in plain text format.

Creating an executable program. After creating and saving the script, type the following command at the DOS prompt:

```
DEBUG < filename
```

where *filename* is the name of the Debug script you created. For example, if the name of your Debug script is KEYPRESS.SCR, you'd type this line:

```
DEBUG < KEYPRESS.SCR
```

at the DOS prompt. The executable program created by Debug will have the extension .COM. The name of the executable file is determined by the contents of the script. Our convention is to use the same name for the executable file as we do for the script. Thus, the executable file created by KEYPRESS-.SCR will be named KEYPRESS.COM. Once you've created the executable file, you run it by typing its name at the DOS prompt. To run KEYPRESS.COM, type KEYPRESS.

PATHS AND THE PATH STATEMENT

DOS World articles often tell you to make sure that a particular file is in a directory included in your PATH statement. This lets you runs a .COM, .EXE, or .BAT file from any directory on any drive.

For example, an author might tell you to create a batch file called TEST.BAT, put it into a subdirectory called \BAT, and put the subdirectory into your PATH statement. You can then execute TEST.BAT by typing TEST from anywhere on your drives, without having to change to the \BAT directory first.

The PATH statement is a line in your AUTOEXEC.BAT file. It gives DOS a list of directories to search for requested files. Here's an example:

```
PATH=C:\DOS;C:\WINDOWS;C:\BAT
```

When you type TEST at the DOS prompt, DOS looks for the program first in the current directory, then in the root directory, and then, in order, the \DOS, \WINDOWS, and \BAT directories. When it finds TEST.BAT in the \BAT directory, it executes the batch file.

BASIC DEFINITIONS

DOS prompt. Also known as the *command prompt.* By default, the DOS prompt looks like this: C:\>. This is where you type the instructions to run programs or DOS commands.

Boot, boot up, reboot. The process of starting or restarting your computer. Turning on your computer is *booting* or *booting up*. Pressing the key combination Ctrl+Alt +Del restarts, or *reboots* your computer. So does pressing the reset button, if your computer has one.

Extensions. When we refer to a program by its common name (for example, the DOS command FORMAT) without an extension, you can assume that the extension is .COM or .EXE. When we refer to a batch file, we always include the extension .BAT. QBasic program names must always include the .BAS extension.

ASCII. American Standard Code for Information Interchange. For our purposes, an ASCII file is a plain text file, one that consists entirely of

the characters you see on your keyboard.

Directories. Your hard drive has a main directory called the *root* or *home* directory. Directories created off the root directory are called *subdirectories*. When we provide the name of a subdirectory, it will look something like this: \WORD\FILES. Here, the root directory has a subdirectory called WORD, which in turn has a subdirectory called FILES.

File placement. We assume that the following files are in your root directory: AUTO-

EXEC.BAT, CONFIG.SYS, and COMMAND.COM. We also assume that your DOS files are in a DOS subdirectory, usually called \DOS.

Keystroke combinations. When you should hold down one key while pressing a second, we indicate it this way: Alt+F4 (press the Alt key and hold it down while you press the F4 key). When you should press one key, release it, and press another, we indicate it this way: Alt, F4 (press the Alt key, release it, then press the F4 key).

DOS WORLD
Q&A

Edited by Doug Lowe

DOS

Q I'm writing a custom installation batch file that makes extensive use of ANSI.SYS. The problem is that if someone uses the program on a system whose CONFIG.SYS file doesn't include a DEVICE=ANSI.SYS command, 90 percent of my work is wasted. The user won't see the colors or the fancy pop-up menus. Can you suggest a way to activate ANSI.SYS after DOS boots so that my batch file will work on systems that don't load ANSI.SYS in CONFIG.SYS?

A The easiest way to solve your problem is to obtain a program that can load an MS-DOS device driver from the command line. Then you can load the ANSI.SYS driver directly from your batch file. Two such programs are DEVADD.COM and DEVLOD.COM. DEVADD.COM is available on the *DOS World* bulletin board (603-924-3181; see the "BBS" section of "How to Use This Magazine," opposite, for dialing instructions). DEVLOD.COM comes with Andrew Schulman's book *Undocumented DOS* (Addison-Wesley, 1993). (If you're interested in the inner workings of DOS, this book is an excellent place to start. It assumes that you have a background in programming, though, so be forewarned.)

If you use DEVADD.COM or DEV-LOD.COM to load ANSI.SYS, you should first make sure that ANSI.SYS isn't loaded into memory already. Adding the following lines to the beginning of your batch file should do the trick:

```
@ECHO OFF
MEM /C | FIND "ANSI"
IF NOT ERRORLEVEL 1 GOTO CONTINUE
loader C:\DOS\ANSI.SYS
:CONTINUE
```

where *loader* is the name of the device-loader program you use: DEVADD.COM or DEVLOD.COM.

This batch-file fragment works by sending the output of a MEM /C command to a FIND command, which looks for the string *ANSI*. The FIND command sets the value of ERROR-LEVEL to 1 if the text isn't found; if the text is found, ERRORLEVEL is set to zero. If ANSI.SYS isn't loaded (the ERRORLEVEL is 1) the subsequent line loads it. If ANSI.SYS is loaded (the ERRORLEVEL is zero), the program skips to the label CONTINUE, and DOS executes the remaining lines in your batch file.

This technique should work most of the time, but it isn't 100-percent reliable. For example, another device driver might include *ANSI* in its name, or the user might have an ANSI.SYS replacement that doesn't include *ANSI* in its name. Another possible problem is that the user might store ANSI.SYS in a directory other than C:\DOS. (For a different way to deal with this problem, see the accompanying tip, "Checking for ANSI.SYS," page 66.)

Q I have several large Lotus 1-2-3 files to which I often add numerous entries. When I copy the expanded files back to my hard disk, I'm certain that they're becoming increasingly fragmented. But I'm afraid to use a defragmenting utility, because I lost some important files when I tried it before. My question is this: Can I defragment these files by saving them under a new name, deleting the old file, and then renaming my new file with the old name?

A Your idea is well intentioned, but it won't work. In fact, taking the approach you suggest may increase the fragmentation on your hard disk.

To understand why, keep in mind that fragmentation isn't a design defect of DOS. Imagine what DOS would have to do to save a 1MB file without fragmentation: search the entire disk until it found an uninterrupted 1MB block of free disk space. If a 1MB block of free space didn't exist, DOS wouldn't be able to store the file at all.

When you copy a file, DOS doesn't worry about how big a space the file needs before deciding where to put it. It places the copy in the first available block of disk space it finds, whether that block is large enough to store the entire file or not. That means that if you copy the file, it, too, may be fragmented. In addition, deleting the old copy of the file will open up a block of disk space, further fragmenting your hard disk.

You can't avoid fragmentation on a DOS system. All you can do is periodically defragment your disks. I hate to be the bearer of bad news, but my only advice is to give defragmenting another try.

Make sure you have a full backup of your hard disk before beginning, and use only a top-quality program, such as MS-DOS 6.x's DEFRAG.EXE or the defragmenting utilities that come with Symantec Corp.'s Norton Utilities or Central Point Software's PC Tools Pro suite.

Q Microsoft's new DEFRAG.EXE program is driving me crazy. When I start it, the disk map indicates that my hard disk is filled with hundreds of "unmovable files." After I defragment the disk, the unmovable files are still there, right where they started, which must mean that my disk is as fragmented as ever. So what exactly is an unmovable file, and how can I thoroughly defragment my hard disk?

A An unmovable file is just what its name suggests: a file you shouldn't move. DEFRAG considers unmovable any file marked with DOS's system attribute. On most computers, only a few files are marked with that attribute—usually just the two system files, MSDOS.SYS and IO.SYS.

To find out which files on your hard disk have this attribute, change to the root directory and type this at the DOS prompt:

```
DIR /AS /S
```

If this command doesn't produce a list of hundreds of unmovable files, your disk probably has lots of "lost clusters": portions that have become detached from the files to which they belong.

DEFRAG treats lost clusters as unmovable files. Try running the SCANDISK command to find and remove these lost clusters. In all likelihood, that will eliminate the unmovable files and permit the DEFRAG utility to do its job.

DOS TIP — CHECKING FOR ANSI.SYS

I enjoy writing batch files for myself and others, and I often embed ANSI.SYS control codes in my programs. The batch files that use these codes produce a garbled mess if the user's CONFIG.SYS file doesn't load ANSI.SYS. To overcome this problem, I add the following snippet to the beginning of my batch files:

```
SET AC=Y
MEM /MANSI > AC.TEMP
FIND /C /I "ANSI is not" AC.TMP > NUL
IF NOT ERRORLEVEL 1 SET AC=N
```

This sets an ANSI.SYS check flag to Y (for *yes*) in the environment. The next line redirects the output of the MEM command to a file called AC.TMP. The switch /MANSI (you may also write it as /M ANSI) tells MEM to report how ANSI.SYS is using memory. If ANSI.SYS is loaded, the message "ANSI is using the following memory" and a rundown on memory use are written to AC.TMP. If ANSI.SYS isn't loaded, the message "ANSI is not currently in memory" is written to AC.TMP. The FIND command searches AC.TMP for "ANSI is not." If it finds this string, it changes the value of the environment variable AC to N.

Further on in the batch file, I add IF statements to determine whether to use ANSI.SYS codes to display text. For example, the following lines display the words *DOS World* in white letters on a red background if ANSI.SYS is loaded:

```
IF(%AC%)==(Y) ECHO ESC[41;37;1mDOS World
IF(%AC%)==(N) ECHO DOS World
```

The program displays the message in white characters on a black background if ANSI.SYS isn't loaded. (You must substitute an escape character—a left-pointing arrow—for ESC in the first command. To do this while working in EDIT.COM, press Ctrl+P and then Ctrl+Esc.)

At the end of the batch file, I clean up by adding these lines, which delete AC.TMP and remove AC from the environment table:

```
ECHO Y| DEL AC.TMP
SET AC=
```

—Glynn Richardson

HARDWARE

Q My mouse is jerky. When I move the pointer across the screen, it often gets stuck in one spot and then seems to jump around. Sometimes, I can't get it to go where I want it to go. Do I need a new mouse?

A Not necessarily. When was the last time you cleaned its innards? Turn your mouse over and look at the ring that holds the ball in place. Give this ring a twist until it comes off, and the ball will fall out. (Catch it if you can.)

Clean off the ball with one of those soft, clean, white cloths that everyone is supposed to have lying around, and then look inside the mouse. If you can see the rollers that detect the ball movement, clean them off, too. Now drop the ball back in and reattach the ring; the mouse should be good as new.

Check your mouse pad, too. If it's covered with grunge, clean it up, or throw it out and get a new one. Believe me, the mouse ball is very good at picking up junk and depositing it inside the mouse. ■

Contributing Editor Doug Lowe is the author of 15 computer books, including The Only DOS Book You'll Ever Need *(Murach),* The Least You Need to Know About DOS *(Murach), and* The Microsoft Press Guide to DoubleSpace *(Microsoft Press).*

UPGRADE UPDATE

■ **EZ-Menu version 6.0**, a system that organizes all DOS and Windows applications into a series of categorized menus, now includes user-defined help boxes, multiuser log-in and security, menu sorting, and network support. Price is $69.95, with special upgrade prices available. *Progressive Computer Services Inc., P.O. Box 7638, Metairie, LA 70010; 504-831-9717; fax 504-834-2160; Internet* ***pcs@f21.n396-.z1.fidonet.org***.

■ New editions of **Opti-Net** and **Opti-Net NLM** networking software, from Reed Technology and Information Services, make it easier for MS-DOS, PC DOS, and Windows users to share CD-ROM drives and multimedia applications. Instead of equipping each workstation with its own CD-ROM drive and disc, you can share multimedia information the same way you access and share files on remote or local hard disks. Prices begin at $795 for the eight-user version. *Reed Technology and Information Services, 20251 Century Blvd., Germantown, MD 20874-1196; 800-922-9204; fax 301-428-0224.*

real world. Let's say you spend hours poring over ads and catalogues in search of the hard disk with the lowest price per megabyte. You set-

Sneaky math: When does 528MB really equal 504MB?

tle on a nice-looking drive described as having a 528MB capacity. But when you get it home and installed, you find that FDISK reports only 504MB. Where did the other 24MB go? The answer is, they were never really there.

Here's what happens. Let's assume your disk contains 1024 cylinders, 16 heads, and 63 sectors per track. All DOS disks use 512 bytes per sector, so the math turns out to be 1024x16x63x512=528,482,304 bytes.

But remember, a million bytes isn't a megabyte. To convert to megabytes, you have to divide by 1,048,578—yielding 504 megabytes.

The problem gets worse with larger drives. A 9-billion-byte drive isn't 9GB, but only 8.4GB—a 7-percent reduction in advertised capacity.

So always ask your dealer for the specific head, cylinder, and sector coordinates, and to avoid getting ripped off, do the math yourself.

—*Robert L. Hummel*

A Winner Over There Débuts Over Here

In Europe, where DOS is still king, Quarterdeck Corp.'s prototype power package for game enthusiasts, **Game-Runner**, has played to rave reviews in recent months. The response has been so favorable that Quarterdeck plans to introduce GameRunner as a DOS and Windows product for the American market early this summer.

For the game devotee, GameRunner could best be described as a frustration-buster utility. Says Quarterdeck's Brad Peppard, it "will answer game players' three most common complaints: out-of-memory errors, having to use boot disks, and enjoyment limitations such as poor scrolling."

GameRunner will include Quarterdeck's renowned memory manager, QEMM; a high-performance cache; and a "tweaking" patch that will make it easier to play and even win any game. The controllable tweaking feature won't take the fun or challenge out of playing; you'll be able to decide whether you need just a tiny clue or the very broadest of hints, or even no help whatsoever. At press time, Quarterdeck (150 Pico Boulevard, Santa Monica, CA 90405; 310-392-9851) was anticipating a May release, with an estimated retail price of less than $50.

—*Steven F. Smith*

Time to Clean House

Dumping all those unwanted DOS and Windows applications is one thing; doing it safely is another matter entirely. **Remove-IT** from Vertisoft Systems not only gets rid of your outdated apps, files, drivers, and fonts with surgical precision, it also scans your hard drive to recommend deletion of duplicate, unused, or unreferenced files. For an encore, Remove-IT automatically logs new software installations, lets you view each file or font before deleting, backs up all

removed applications to either your hard disk or a floppy, and features instant undo of your most recent deletion. It also finds time to create a bootable recovery disk in the event of a corrupted configuration file or CMOS setup.

System requirements are a 386 PC or later, DOS 4.0 or later or Windows 3.1 or later, and 2MB of memory. Suggested price is $69.95. Contact Vertisoft Systems, 4

Embarcadero Center, Suite 3470, San Francisco, CA 94111; 800-466-5875; fax 800-466-4719.

If you're a Word-Perfect fan, **Flash Forward for Word-Perfect DOS and WordPerfect Windows**, from Summit Software Inc., can speed up production of frequently used phrases and form documents.

Targeted especially at medical transcriptionists, court reporters, and legal secretaries, Flash Forward lets you abbreviate words, phrases, sentences, redundant sequences, and, in the Windows version, entire documents of up to 64,000 characters—then expands each abbreviation to the desired term or phrase.

For example, you can set up a form letter, type *sy*, and Flash Forward will

A Caffeine Buzz from WordPerfect?

immediately expand to *Sincerely yours, plus individual's name and title*.

Flash Forward is similar to a WordPerfect macro, in that it automates repetitive words and phrases. The difference, claims Summit's Greg Walberg, is that Flash Forward leaves macros in the dust: "Macros are used by almost no one because they're so cumbersome [and] confusing, and [they] don't really save time."

Included as special features are an on-screen prompt to remind you if you

forget to use a short version of a word or phrase, and an optional audio reminder for unabashed "hunt and peck" types.

The $80 program comes with 22 short forms of commonly used words; you can then build a list of as many abbreviations as you like. You can also shorten names of months and common word endings such as *-ing*, *-tion*, and *-able*. As Walberg puts it, "Flash Forward speeds up Word-Perfect about the same way a triple cappuccino speeds up the day."

To see for yourself, contact Summit Software at 46750 Fremont Boulevard, Suite 207, Fremont, CA 94538; 510-770-3495.

As this issue of **DOS World** was going to press, CyberMedia was getting ready to throw out a life line to its own rescue utility, **PC911**. The product was originally designed to rescue a PC from common problems such as IRQ and DMA conflicts, as well as loss of system information when a computer freezes or can't boot up. Version 2.0 will address concerns raised by users of the original PC911, now automatically testing its emergency boot disk during installation to make sure it

Rescuing A Rescuer

has been created properly, supporting large-capacity drives that use OnTrack's Disk Manager, allowing for deletion of unwanted system-parameter snapshots, and presenting a simpler interface. Commercial release was expected after April 15, with an upgrade of the new version shipped free of charge to all registered PC911 version 1.0 users. Contact CyberMedia, 1800 Century Park East, Suite 1145, Century City, CA 90067, 310-843-0800.

Attitude Is Everything

The Mental Edge series is a new collection of DOS and Windows self-improvement programs, available for only $10 each, from Mind Media (formerly Mindware) of Santa Cruz, California.

DOS titles include IQ builders called Turbocharge

Your Brain, Motivation Advisor, Brainstorm Idea-Gen, ThunderThought Creativity Pack, and Career Path.

Windows users can choose from Achievement Planner, PC Therapist, and Subliminal Messenger.

Products like these normally

retail for up to $195, so Mind Media's offer is quite a bargain. You can get a complete listing of Mind Media programs by calling 408-426-0762 (fax 408-426-8519), or write to Mind Media at 349 Almar Avenue, #C1125, Santa Cruz, CA 95060.

■ **Freight Train** is the newest DOS simulation software game for railroad hobbyists, published by Abracadata of Eugene, Oregon. Building on RR Switch, the company's previous railroad game, Freight Train features full-color graphics, thousands of playing scenarios, and Sound Blaster compatibility. The object of the game is to successfully complete the complicated work of running a short-line freight operation over a simulated 12-hour period. Price is $39.95. *Abracadata, P.O. Box 2440, Eugene, OR 97402; 503-343-2324; fax 503-343-2450.*

■ The May 1995 issue of *DOS World* reviewed **Buttons for DOS version 5.0** (page 68), a program that pops up on exit from Windows to give you point-and-click mouse access to all your DOS games and applications. Now you can get Buttons in shareware format. Requires a VGA monitor and DOS 3.3 or later. Download it from the Triad Software File Area on the Software Creations BBS (508-365-2359), or contact *Triad Software, P.O. Box 1299, Sequim, WA 98382; 800-683-3202.*

■ **Multiuser DOS 7.0 Gold** from Concurrent Controls Inc. lets each station on a network run up to 32 applications simultaneously. Version 7.0 includes expanded memory management capabilities, enhanced remote access, integration of job/program scheduling, e-mail, and disk caching. Call or write for prices. *Concurrent Controls Inc., 880 Dubuque Ave., South San Francisco, CA 94080; 415-873-6091; fax 415-873-6091.*

Parents and teachers of children and teenagers may be relieved by this development: Solid Oak Software has introduced a product called **PG13**, a DOS and Windows utility designed to prevent kids of all ages from gaining access to explicit pictures now available on line, on disk, or on CD-ROM.

Comparable to 900-number blocking services currently offered by telephone companies, PG13 is a completely invisible operation that makes it impossible for

Not for Young Eyes

anyone using your computer to view GIF files.

Flexibility is built in: You can disengage PG13 by entering a simple command, but an unsuspecting user can't disable it.

PG13 retails for $19.95 and works with all DOS and Windows communications and graphics programs.

For more information on PG13, contact Solid Oak Software Inc., P.O. Box 6826, Santa Barbara, CA 93160; 800-388-2761; fax 805-967-1614.

The All-Everything Board

What if you could dial a bulletin-board system, request an on-line report, download it directly to your computer, and then send data by modem and attach voice messages for explanation or comment? Yes, but all from one board?

Best Data Products' new internal **Advanced Communication Enhancement (ACE)** card for OS/2 and Windows is a four-in-one multimedia communications board, merging fax, modem, voice, and sound capabilities with an interface for a Panasonic CD-ROM drive. How does it do all that? The concept capitalizes on IBM's Mwave technology, an advanced digital-signal process (DSP) that allows for complete software upgrades (as for a V.34 modem or a full-duplex speaker phone) and simultaneous tasking (such as typing one document while faxing another).

Specifically, ACE is a 14,400-bps modem with V.42bis for throughput of up to 57,600 bps, and is also a 16-bit audio board with wave-table synthesis. Minimum system requirements are a 386SX 16MHz processor and 6MB of RAM. Suggested price is $299. Check it out by contacting Best Data Products Inc., 21800 Nordhoff Street, Chatsworth, CA 91311; 818-773-9600; fax 818-773-9619.

PC DOS 7

Keep the Flame Burning

IBM's improved memory management, disk compression, and Plug and Play support can revive DOS computing.

by Robert L. Hummel

Microsoft has no DOS upgrade in the works; Novell has decided not to further enhance its DOS. Just when it appeared that DOS would languish, IBM has introduced PC DOS 7 to compete for the heart and soul of the 50-million-plus DOS-upgrade market.

And compete it does. PC DOS 7 provides significant enhancements to memory management, disk compression, and usability. No matter what DOS version—MS-DOS 3.3 or later, PC DOS, or Novell DOS—or Windows you already use, you can upgrade to PC DOS 7 without trouble or conflict.

The biggest news about PC DOS 7 is memory—both freeing it and managing it. Every part of the operating system has been rewritten to consume less conventional memory and to take advantage of a new upper-memory area (UMA). For example, a new CONFIG.SYS command, DOSDATA, loads many of DOS's internal data structures into the UMA. These changes easily free up about 10K of conventional memory on a bare-bones system. If you've been struggling to cram in network drivers, CD-ROM drivers, and mem-

ory resident programs, and *still* need more memory for applications, PC DOS 7 may be your salvation.

Beyond memory improvements, PC DOS 7 addresses concerns over the reliability of DoubleSpace (MS-DOS 6.0's disk-compression utility), and Microsoft's subsequent change to DriveSpace, by bundling Stacker 4.0—considered the premier disk-compression program on the market. Providing Windows and DOS interfaces for Stacker, PC DOS 7 lets you maintain and manage your drives regardless of what operating system you use. Stacker 4.0 can produce portable compressed floppies, create compressed drives as large as 2GB (compared with a 512MB limit under MS-DOS), and provide average compression ratios as high as 2.5 to 1.

PC DOS 7 also promotes REXX (*Restructured Extended Executor*), a programming language that combines simplicity and power. It's quite easy to learn

because it uses familiar words and concepts. Although it doesn't replace DOS's batch language, its capabilities are superior.

Although putting REXX into PC DOS 7 is a great move, IBM has cheated users by neglecting to include any printed documentation, quick-reference card, tutorial, or examples. If you plan to learn about REXX from the ground up, you'll certainly need additional documentation.

New commands in PC DOS 7 include DYNALOAD, to load device drivers from the command line, and CRC (Cyclical Redundancy Checksum), which identifies and troubleshoots a file. A FILEUP program, although awkward to use, lets you synchronize files between two PCs, a PC and a network, or two locations on the same PC system. Support for warm and hot docking of portable computers is also new in IBM's PC DOS 7—a foreshadowing of support for the Plug and Play standard.

Many PC DOS staples have been improved, too. IBM's Antivirus program, for example, has been expanded to handle more than 2100 viruses. Also improved are Phoenix Technology's PCMCIA support and Central Point's Backup, File Undelete, and RAM-Boost memory optimizer.

Although it's impressive, PC DOS 7 isn't without its shortcomings. Compared with PC DOS 6.3, the documentation is paltry. And PC DOS's E-editor, though dramatically improved over earlier versions, is in no danger of losing its unenviable position as the worst text editor ever written.
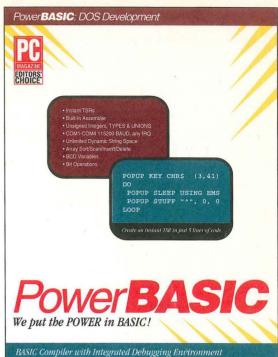
Overall, PC DOS 7 represents a significant advancement of the DOS standard. Taken individually, the improvements in disk compression and memory management and the inclusion of REXX alone justify the upgrade. Add the enhancements to the operating system's other utilities, and PC DOS 7 is an unbeatable bargain.

Robert L. Hummel is a journalist, programmer, and engineer. He's a contributing editor and feature writer for DOS World *and also serves as Windows editor, columnist, and feature writer for* Maximize *magazine.*

NEOPAINT PRESENTATION PACK

A Multimedia Marvel
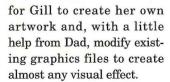
"So easy and yet so stunning" only begins to describe this DOS-based artistic wonder.

by Stanley J. Wszola

The ultimate test of any software package is to try to do real work with it. My daughter, Gill, needed images of planets and galaxies for her astronomy-class poster, so it was an ideal opportunity for me to unwrap a freshly obtained copy of NeoPaint Presentation Pack, from NeoSoft Corporation—a DOS-based paint, image-editing, and presentation studio. Could it rummage through my graphics files and help Gill complete her assignment with a bit of panache? Could it ever!

Presentation Pack (a combination of NeoPaint and the multimedia program Neo-Show) features an outstanding set of drawing tools that make it easy to create anything from simple lines to blended colors, or transparent overlays to acid wash. You can create illustrations for desktop-publishing projects, edit digitized photos, or just have a blast doodling. Neither of us is an artist, but NeoPaint made it easy

for Gill to create her own artwork and, with a little help from Dad, modify existing graphics files to create almost any visual effect.

It took only a few minutes to explain to Gill the ins and outs of NeoPaint. Friendly pull-down menus make it easy for novice or expert to start a new file, edit, or modify any picture.

It didn't take the two of us very long to find several files of depictions of galaxies and copy them to a separate subdirectory. When we called them up, the full-screen display was dramatic. Gill couldn't wait to print them, but I was concerned that printing a picture of a spiral galaxy against the blackness of space would exhaust my print cartridge. Fortunately, NeoPaint had a solution: the Invert tool, which we used to create a stunning poster. We reversed the black and white so that we had a *black* galaxy on a *white* background.

Because NeoPaint comes with a variety of fonts (additonal typefaces are also available), Gill was able to make professional-looking headlines and subheads for her poster. She also decided to enclose the text in some of NeoPaint's built-in shapes and boxes.

But don't be mistaken— NeoPaint Presentation Pack isn't just for kids. Although my daughter didn't need the NeoShow multimedia program for her poster project, I found it to be just as impressive and fun to use as the paint module. In fact, Neo-Show is an ideal business tool, containing a number of visual-effects aids that let you combine PCX and GIF images with audio files for computer-based presentations and audiovisual displays. For that professional look that will wow coworkers and clients alike, you can add fades, wipes, and animated special effects.

Depending on your video card, NeoPaint can change the screen resolution and colors to match the resolution and colors of a graphics file. Each graphics file can always be displayed full size in the editing window with the correct color palette.

Although the package includes PIF and icon files,

beware of running NeoPaint Presentation Pack under Windows. Changing video modes trashed my display, and the combination of Windows and the paint module slowed the performance of my computer to unusable levels.

NeoPaint Presentation Pack runs on any PC, from an 8088 to a Pentium, but a 286 or faster CPU is recommended. It requires 1.5MB of hard-disk space and 640K of system RAM. It supports expanded (EMS), extended (XMS) and virtual-disk memory for working with large images, and requires a Hercules monochrome, EGA, VGA, or SVGA video card. An SVGA card with sufficient memory is recommended for higher resolutions.
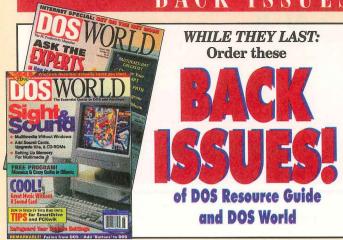
You'll also need a mouse or other pointing device; be sure to load your DOS mouse driver (MOUSE.COM) before you get started.

Stanley J. Wszola is a freelance writer and systems-support analyst. Contact him on CompuServe at 71011,1726.

DOS Watch

News and reviews of DOS enhancements, upgrades, and products

Edited by Steven F. Smith

A recent Software Publishers Association (SPA) survey indicates that although 1994 was a banner year for increased Windows interest, 69 percent of all non-Windows users still say they won't switch from DOS. "I read that as a lot of unhappy people if Microsoft doesn't deliver an upgrade to DOS, independent of Windows 95, considering MS-DOS is the most-used operating system throughout the world," says Connecticut software developer J.C. Bedell.

But don't hold your breath. Microsoft remains firmly noncommittal on the prospect of MS-DOS 7. A year ago, executives in Redmond hinted that they would offer a separate DOS 7 only if they perceived a sufficient level of interest following the launch of Windows 95.

But that interest already does exist, claims David Pacheco, a DOS forum assistant on America Online: "There are thousands of corporate sites that will not upgrade to Windows [95] for quite some time, particularly with the additional resources required. A DOS 7 upgrade from Microsoft at a small cost would be something to keep users happy."

If Microsoft Is the Question, Is IBM the Answer?

Meanwhile, could it be IBM to the rescue? Big Blue is hoping to capitalize with its just-released PC DOS 7. It features what some experts call the best integrated version of Stacker disk compression and a smaller conventional memory requirement. (See our review, page 74.) According to *PC World*, IBM's venture may be the only upgrade path left for DOS devotees. *—Kate Chase*

B E W A R E
The Power Of Two

Psst . . . . Want nothing for something? It can happen when you buy a hard disk these days. But forewarned is forearmed: Here's how to avoid getting ripped off in one of the computer industry's biggest (and perhaps least publicized) scams.

In the world of computers, the number 2 is king. Memory size, disk space, and everything else are expressed in terms of 2 or a power of 2. For example, a kilobyte is 2^{10} (2 to the tenth power), or 1024, bytes. A megabyte is 2^{10}, or 1024, kilobytes—or 1,048,576 bytes. And a gigabyte is 2^{10}, or 1024, megabytes, or 1,048,576 kilobytes, or 1,073,741,824 bytes.

Still with us? The math is straightforward when you're dealing with computers, but trouble erupts when the world of silicon meets the

FROM THE BOOKSHELF: ALL IN A DAY'S WORK, SORT OF

If you've got the time, you can conquer the ins and outs of DOS in a matter of hours (weeks? months?) with *DOS in a Day*, an exhaustive yet down-to-earth book by Jeff Weber of Weber Systems, Chesterland, Ohio. With more than 600 pages of hands-on tutorials and study tips, however, the title shouldn't be taken too literally. There's plenty to discover: everything from the ABCs of computing, to disaster recovery and memory management, to maximizing performance.
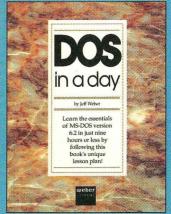
Weber's goal is to present only those DOS topics required to manage programs and information effectively and to ignore those obscure, technical features of DOS that are of interest only to the programmer or advanced user. The book is designed around practical lessons, each chapter concluding with true-false and multiple-choice "test" questions for personal or classroom use.

Then, once you've mastered DOS, you can go on to the companion volume, *Windows in a Day*. Weber's books are priced at $29.95 each, from Weber Systems, 8437 Mayfield Road, Chesterland, OH 44026; 216-729-2858; fax 216-729-3203.

DOS in a day
by Jeff Weber
Learn the essentials of MS-DOS version 6.2 in just nine hours or less by following this book's unique lesson plan!